

IBM i
7.3

Connecting to IBM i
IBM i Access Client Solutions



Note

Before using this information and the product it supports, read the information in [“Notices” on page 107.](#)

This edition applies to IBM i 7.3 (product number 5770-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

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

© **Copyright International Business Machines Corporation 2013, 2021.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

IBM i Access Client Solutions.....	1
What's new.....	1
PDF file for IBM i Access Client Solutions.....	1
Overview of IBM i Access Client Solutions.....	1
Getting started with IBM i Access Client Solutions.....	2
License information.....	2
Getting product updates.....	4
IBM i Access Client Solutions console connections.....	4
Related information for IBM i Access Client Solutions.....	4
Setting up the IBM i platform.....	5
Required and optional programs to install.....	5
Configuring TCP/IP on IBM i platform.....	6
Configuring IBM i NetServer on IBM i.....	6
Host server administration.....	7
Identify IBM i host servers and associated programs.....	7
Host servers by function.....	8
File server.....	9
File server programs.....	9
Database server.....	10
Data queue server.....	15
Network print server.....	15
Central server.....	16
Remote command and distributed program call server.....	16
Signon server.....	17
Server Port Mapper.....	17
Extended Dynamic Remote SQL server (QXDAEDRSQL).....	17
DRDA/DDM server.....	17
Use IBM i host servers.....	18
Establish client/server communications.....	18
IBM i Subsystems.....	24
IBM i system values.....	35
Identify and display IBM i server jobs.....	38
Use server exit programs.....	40
Register exit programs.....	41
Write exit programs.....	43
Exit program parameters.....	43
Examples: Exit programs.....	64
Console Advanced Topics.....	78
Console planning considerations.....	78
Planning considerations for your backup console.....	80
Console hardware requirements.....	81
Planning considerations for your Operations Console installation or upgrade.....	81
Preparation for your network environment.....	82
Security of your Operations Console configuration.....	82
Scenario: Understanding your configuration.....	83
Scenario: Consoles for multiple systems or partitions.....	84
Managing Operations Console.....	84
Takeover or recovery of an Operations Console connection.....	84
Takeover details.....	85
Recovery details.....	85
Enabling console takeover.....	86

Scenario: Takeover and recovery.....	87
Management of multiple consoles.....	87
Multiple PC consoles on a network.....	88
Changing from one console type to another.....	88
Switching from one console type to another when the current console is not operational...	88
Managing your local console on a network.....	89
Starting the system using a manual IPL.....	89
Using the operations console service functions.....	90
Using the OPSCONSOLE macro.....	95
Changing the value of the display console status screen option.....	96
Changing the value of the Console F18 takeover option.....	96
Configuring a service host name (interface name).....	97
Deallocating or moving the LAN adapter card from use by Operations Console.....	98
Changing network values for Operations Console (LAN).....	99
Troubleshooting Operations Console connection.....	100
Troubleshooting connection problems.....	100
Local console connection problems.....	100
Troubleshooting system reference code data.....	101
System reference code A6nn500x.....	101
System reference code A6005001, A6005004, and A6005007.....	101
System reference code A6005008.....	102
System reference code A6005082.....	104
System reference code A9002000.....	104
IPL step C6004031 takes longer than expected.....	104
Troubleshooting other Operations Console problems.....	104
System requests do not work.....	104
Unable to sign on because of a lost or expired password or disabled user ID.....	105
Notices.....	107
Programming interface information.....	108
Trademarks.....	108
Terms and conditions.....	109

IBM i Access Client Solutions

IBM i Access Client Solutions is the newest member of the IBM i Access Family of products. It provides a platform-independent solution that runs on most operating systems that support Java™, including Linux®, macOS, Windows, and IBM i. IBM i Access Client Solutions consolidates the most commonly used tasks for accessing your IBM i into one simplified location.

What's new

Read about new or significantly changed information for the IBM i Access Client Solutions topic collection.

IBM i Access Client Solutions is the newest member of the IBM i Access Family of products for connecting to your IBM i. IBM i Access Client Solutions replaces IBM i Access for Windows for connecting to your IBM i.

How to see what's new or changed

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

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

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

To find other information about what's new or changed this release, see the [Memo to Users](#).

PDF file for IBM i Access Client Solutions

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select [IBM i Access Client Solutions](#).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
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**.

Overview of IBM i Access Client Solutions

IBM i Access Client Solutions is a key offering in the IBM i Access Family. It offers a powerful set of capabilities for connecting PCs to IBM i.

IBM i Access Client Solutions provides a Java-based platform-independent interface, which runs on most operating systems that support Java including Linux, macOS, and Windows. IBM i Access Client Solutions consolidates the most commonly used tasks for managing your IBM i into one simplified location.

IBM i Access Client Solutions uses the same IBM i host servers as the other IBM i Access family products and requires the same IBM i Access Family license (5770-XW1) to use the 5250 emulation and Data Transfer features.

IBM i Access Client Solutions includes the following features.

- Full featured 5250 display emulator that is based on IBM Rational® Host On-Demand. In addition to all the 5250 display features you are accustomed to when you use IBM i Access for Windows, you can now switch your 5250 display emulator between languages without rebooting your workstation. In addition,

you can have multiple concurrent sessions with different host code pages. This feature allows separate languages to be displayed within different emulator sessions. Printer emulation is also supported

- 5250 Session Manager that is modeled after IBM Personal Communications Session Manager can be used for managing your 5250 emulator sessions
- Data Transfer allows you to transfer data from/to your IBM i database to/from various file types on your workstation such as OpenDocument spreadsheet (*.ods), Excel Workbook (*.xlsx), and other file formats
- Printer Output provides an interface to view files in the IBM i output queues and provides the capability to download these files to your client system
- Virtual Control Panel with a graphical interface to the IBM i operation panel
- 5250 emulation for LAN Console
- Consolidation for hardware management interface configurations that include ASMI, IVM, and HMC
- launch capability to IBM Navigator for i using your default browser

IBM i Access Client Solutions includes optional platform-specific packages for Windows, Linux, macOS, and PASE. These packages provide middleware for using and developing client applications that run natively on those operating systems as well as provide database access technologies.

Related information

[IBM i Access Client Solutions: Application Packages](#)

Getting started with IBM i Access Client Solutions

Before you begin using IBM i Access Client Solutions you must deploy the product.

You can acquire IBM i Access Client Solutions by downloading it from the Entitled Software Support (ESS) website.

You can acquire media by ordering a refresh feature. The physical media for IBM i Access Client Solutions does not contain the optional Windows and Linux Application Packages. Those packages are only available from the Entitled Software Support (ESS) website.

1. Download the zip archive file for IBM i Access Client Solutions. The IBM i Access Client Solutions web site gives details of how to obtain the IBM i Access Client Solutions archive file.
2. Extract the contents of the archive file.

More details can be found in the QuickStartGuide and GettingStarted documents that are included in the product's Documentation directory.

Prerequisites

IBM i Access Client Solutions runs on most operating systems that support Java 8.0 or higher including various versions of Linux, Mac, and Windows.

Related information

[Entitled Software Support](#)

[IBM i Access Client Solutions web site](#)

License information for IBM i Access

IBM i Access Client Solutions is a licensed program. Some features require an IBM i Access Family (5770-XW1) license before you can use them.

To use the IBM i Access Family licensed program (5770-XW1), you must determine the usage limit of the license, update the usage limit on your IBM i platform, and enter the license key information.

The following features require an IBM i Access Family license and an IBM i license before you can use them:

- 5250 Display and Print Emulator

- Data Transfer
- IBM i Access for Web

Important: A software license key is required for IBM i Access Family 5770-XW1. IBM i Access Family is included on the Keyed Stamped Media that comes with all IBM i software orders for the specific release. You receive a license key if you order 5770-XW1. If you have not ordered 5770-XW1, you can evaluate the product from the keyed stamped media for 70 days. At the end of the 70-day evaluation period, the product will be disabled if you have not ordered the product and received a software license key. The software license key is an 18-digit authorization code that allows the software product and feature on the keyed stamped media to be used on a specified IBM i platform.

All features other than 5250 Display and Print Emulator and Data Transfer features require only an IBM i license before you can use them.

Usage limit for a license

IBM i Access Client Solutions clients are licensed by the number of concurrently active PCs accessing IBM i platforms. A PC running IBM i Access Client Solutions holds a license through the duration of the licensed function plus additional time that is specified in the IBM i Access Client Solutions Preferences panel. When this time expires, the license is available for another PC to use. If a PC accesses a licensed program on more than one IBM i platform, that PC requires a license on each IBM i platform to which it connects using a licensed function.

When using IBM i Access Client Solutions on a PC, more than one session to the IBM i platform can be established on that PC, but only one IBM i Access Family license is used. For example, you can start many 5250 emulation or Data Transfer sessions, but the PC requires only one license.

IBM i Access for Web (5770-XH2) is another product that requires IBM i Access Family licenses. Be aware that if the IBM i Access for Web product is being used on the same PC as IBM i Access Client Solutions, each of those products will use a separate license. Therefore, when using both products, one PC will use a minimum of two IBM i Access Family licenses. For more information about license usage for IBM i Access for Web, see the [License information for IBM i Access for Web](#) topic.

Licensing is managed at the IBM i Access Family level. Therefore, any combination of the IBM i Access Client Solutions and IBM i Access for Web clients is allowable up to the license limit.

To determine the IBM i Access license usage limit, do the following:

1. Type the WRKLICINF command on the IBM i platform to which you intend to connect. A list of products appears.
2. Type 5 in the entry field next to the product 5770-XW1, Base, Feature 5050. This will display the details for the IBM i Access Family License product, including the usage limit. The usage limit should be equal to the number of licenses that are purchased for IBM i Access Family. Any number exceeding the purchased limit violates the IBM license agreement.

Use the IBM i Access Family licensed program

To use the IBM i Access Family licensed program (5770-XW1), you must update the usage limit and enter the license key information.

1. To update the usage limit for the 5770-XW1 product on your system, do the following:
 - a. Type the WRKLICINF command on the IBM i platform to which you intend to connect. A list of products appears.
 - b. Type 2 in the entry field next to the product 5770-XW1 Base, Feature 5050. Change the usage limit to the number of licenses that you have purchased for IBM i Access. If you have purchased the processor-based option for IBM i Access, enter the value *NOMAX for usage limit. Entering any number that exceeds the purchased limit violates the IBM license agreement.
2. To enter the license key information, do the following:
 - a. Type the WRKLICINF command on the IBM i platform to which you intend to connect. A list of products appears.
 - b. Type 1 in the entry field next to the product 5770-XW1 Option 1, Feature 5101.

c. Enter the license key information.

Notes: ADDLICKEY values for Usage Limit and Processor Group for Feature 5101:

- Always enter the value *NOMAX in the **Usage Limit** field for Feature 5101. The usage limit value *NOMAX is required as part of the software key. The value *NOMAX is not a representation of the number of users licensed under a user-based license.
- For a user-based license, enter the value *ANY in the **Processor Group** field. A processor-based license has a processor group value that corresponds to the processor group licensed.

For information about entering license key information, see [“Required and optional programs to install” on page 5](#).

Note: After installing the 5770-XW1 product and entering the license key information, the Central server jobs (QZSCSRVS) must be restarted. Subsequent license requests might fail if the restart is not done. The QZSCSRVS job can be controlled with the Start Prestart Job (STRPJ) and End Prestart Job (ENDPJ) commands. The IBM i commands to end and restart the Central server jobs (QZSCSRVS) running in the QUSRWRK subsystem:

1. ENDPJ SBS(QUSRWRK) PGM(QSYS/QZSCSRVS) OPTION(*IMMED)
2. STRPJ SBS(QUSRWRK) PGM(QSYS/QZSCSRVS)

Getting product updates

You can obtain updates for IBM i Access Client Solutions from the same location where you downloaded the product. Just download the product again to receive the latest version.

You can acquire IBM i Access Client Solutions by downloading it from the Entitled Software Support (ESS) website.

For more information about downloading the product, go to the IBM i Access Client Solutions web site.

Related information

[Entitled Software Support](#)

[IBM i Access Client Solutions web site](#)

IBM i Access Client Solutions console connections

IBM i Access Client Solutions uses a user ID and password (provided by the user) as part of an TLS handshake when connecting to an IBM i console.

The user ID must have operator panel privilege to connect to the Virtual Control Panel. By default, the IBM i provided service tools ID of 11111111 (eight 1's) does not have operator panel privilege. If the service tools ID of 11111111 (eight 1's) is used for connecting to the Virtual Control Panel, its privilege needs to be updated using either SST or DST

Related information for IBM i Access Client Solutions

Web sites and other information center topic collections contain information that relates to the IBM i Access Client Solutions topic collection. You can view or print any of the PDF files.

Web sites

IBM home page

Visit this site for information about IBM products, services, support, and downloads.

IBM i home page

Learn about the IBM i platform.

IBM i Access Client Solutions home page

Visit this site to learn more about IBM i Access Client Solutions.

IBM i Access home page

This Web site includes online product information about IBM i Access.

Navigator for i home page

System i® Navigator is a powerful graphical interface for Windows clients.

IBM software home page

Use this site as a resource for IBM software, trials and betas, software news, information about buying software, and software support.

Support for IBM i

Technical support and resources for IBM i and IBM i.

IBM Redbooks® home page

See this site for additional skills, technical know-how, and materials.

IBM i NetServer home page

See this site for information about IBM i NetServer.

Information APARs

An information authorized program analysis report (Information APAR) is an electronic document that is used to communicate information not found in publications, online information, critical fix information, or other sources.

Setting up the IBM i Platform

Use this information to guide you in setting up the IBM i platform.

Required and optional programs to install

There are programs you should install, depending on the features you will use.

Required licensed program options

<i>Table 1. Required programs to install for IBM i Access</i>		
Program	Option	Description
5770-SS1	12	Host Servers
5770-XW1	Base, 1	IBM i Access Family
5770-TC1		TCP/IP Utilities

Note: Each product that you install needs to be at the latest level.

You only need to install IBM i Access Family 5770-XW1 Base and Option 1 if you want to use Data Transfer, or Display and Printer Emulation, or IBM i Access for Web.

For information about license and usage limit requirements, see the [License information for IBM i Access](#) topic.

Optional licensed programs

To use Secure Sockets Layer (SSL), order and install the products listed below. For these products, you are responsible for making sure that you are using the correct encryption for your region and for the regions in which your IBM i platform does business.

- 5770-SS1 - Digital Certificate Manager (Boss Option 34)
- 5770-DG1 - (HTTP Server)

To take advantage of AFP capabilities when viewing and downloading spool files, install:

- 5770-TS1 - IBM Transform Services for i (*BASE)
- 5770-TS1 - Transforms AFP to PDF (Option 1)

Configuring TCP/IP on IBM i platform

TCP/IP must be configured appropriately, depending on how you will connect the PC to the IBM i platform.

TCP/IP is a licensed program that is shipped with IBM i. This information assumes that you have TCP/IP installed on your IBM i platform. You can find the procedure for installing TCP/IP (5770-TC1) on your system in [Installing additional licensed programs](#).

If you have TCP/IP already set up on your system, then you do not need to perform any additional TCP/IP configuration for IBM i Access Client Solutions.

Note: VPN is an option for secure remote connections. For IBM i VPN information, see [Virtual private networking](#).

Configure TCP/IP for LAN use

If you plan on using IBM i Access Client Solutions over a LAN, then you must configure TCP/IP for LAN use. For information about configuring TCP/IP on your IBM i platform, see [TCP/IP Setup](#).

Configure TCP/IP for PPP or SLIP connections

If you are using SLIP or PPP to connect the PC to the IBM i platform, see the topic about [PPP connections](#) for information about configuring point-to-point TCP/IP.

Related information

[Virtual Private Networking](#)

Configuring IBM i NetServer on IBM i

You can use IBM i NetServer to make IBM i available to PC users.

Note:

- IBM i NetServer allows a Kerberos ticket for user authentication. Kerberos is a third-party authentication mechanism where the client proves its identity to a Kerberos server (or Key Distribution Center), and then receives a ticket in return. The client can then use that ticket to cryptographically prove its identity to other systems on the network. The Kerberos ticket is used to authenticate a user to a system rather than passing user ID and password as the authentication data. Microsoft includes Kerberos authentication support in Windows XP and newer operating systems.

For more information about Kerberos tickets, see [Network authentication service protocols](#).

- To configure IBM i NetServer file and print sharing capabilities, see these instructions:
 - [IBM i NetServer file shares](#).
 - [IBM i NetServer print shares](#).

Additional configuration instructions are available at [Getting started with IBM i NetServer](#).

Configure IBM i NetServer

The following IBM i configuration is necessary if users in your network will be using file and print sharing.

Note:

- The following instructions assume that you have TCP/IP installed and configured on your IBM i platform. If you do not, follow the instructions in [“Configuring TCP/IP on IBM i platform”](#) on page 6.
- The following configuration instructions require you to already have access to a PC that has System i Navigator installed.

To configure your IBM i for IBM i NetServer support with IBM Navigator for i, do the following:

1. Use the IBM i NetServer wizard.

To open the wizard, follow these steps:

- a) Expand **Network** from your IBM Navigator for i window.
 - b) Expand **Servers**.
 - c) Click on **TCP/IP**.
 - d) Right-click **IBM i NetServer** and click on **Configuration**.
2. Follow the prompts provided by the wizard.
Note: For additional information about IBM i NetServer configuration, see [IBM i NetServer](#).
 3. For easier management and resolution of TCP/IP addresses, add an entry for the IBM i NetServer to a Domain Name Server (DNS).
Note: Configuration instructions are located in the IBM Navigator for i online help and [Configuring and connecting your PC client](#).
 4. Changes made to your IBM i NetServer properties do not take effect until the next time IBM i NetServer is started. To start or stop IBM i NetServer:
 - a) Expand **Network** from your IBM Navigator for i window.
 - b) Expand **Servers**.
 - c) Click on **TCP/IP**.
 - d) Right-click **IBM i NetServer** and click on **Start** or **Stop**.

Host server administration

Identify and effectively use and manage IBM i Access Client Solutions host servers.

This topic provides brief descriptions and technical information for a subset of host server functions used by the IBM i Access Client Solutions product.

IBM i host servers

Host servers handle requests from client PCs or devices such as running an application, querying a database, printing a document, or even performing a backup or recovery procedure. IBM i computers are full-function servers capable of performing many tasks at once, including file, database, applications, mail, print, fax, and wireless communications. When these tasks are handled by several different servers, server management and coordination become complex. Having all your servers on one integrated system greatly reduces the overall cost and complexity of managing your network.

These servers are used by IBM i Access Client Solutions but are designed so that other client products can also use them. This topic focuses on how these servers are used by IBM i Access Client Solutions.

Adding or removing the Host Server option

The servers discussed here are all optimized servers, and are included with the base option of IBM i.

If you are not using any IBM i Access products or IBM i NetServer and would like to remove the Host Server option, you should end the subsystems used by these servers before you remove the option. End the QBASE or QCMN subsystem (for host servers with APPC support), the QSYSWRK and QUSRWRK subsystems (for host servers with sockets support), and the QSERVER subsystem (for database and file server). Problems may occur if you try to delete the option while any of these subsystems are active.

Identify IBM i host servers and associated programs.

There are several host servers and related objects that are common for the IBM i Access family.

This information does not include all IBM i host servers. It covers only those used by the IBM i Access family, many of which, are viewable by their type or function.

Host servers by function

IBM i host servers are listed by their associated function.

The following table shows a subset of host servers on the system and that are used by the product.

Client function	IBM i server used
.NET Data Provider	<ul style="list-style-type: none"> • Database Server • Signon server • Central server
IBM Toolbox for Java	<ul style="list-style-type: none"> • Signon server • Central server • File server • Database Server • DRDA and DDM server • Data queue server • Remote command and distributed program call server • Network print server
Data Transfer	<ul style="list-style-type: none"> • Signon server • Central server • Database server
ODBC driver	<ul style="list-style-type: none"> • Signon server • Central server • Database server
Access integrated file system from System i Navigator	File server
Data queue APIs	Data queue server
OLE DB provider	<ul style="list-style-type: none"> • Data queue server • Database server • Remote command and distributed program call server • Central server • Signon server
Extended Dynamic Remote SQL server (QXDAEDRSQL)	<ul style="list-style-type: none"> • Signon server • Central server • QXDAEDRSQL server
License management Done when an application that requires a license is started (Data Transfer and 5250 emulation)	Central server

Client function	IBM i server used
Retrieve conversion map Done only if the client does not contain the required conversion maps	Central server
Remote command functions	Remote command and distributed program call server
Distributed program call	Remote command and distributed program call server
Send password for validation and change expired password	Signon server
Network Print	Network print server

File server

The IBM i file server and its associated programs work with the integrated file system.

The IBM i architecture supports several different file systems with similar interfaces. The integrated file system is a part of the base IBM i operating system that supports stream input/output and storage management, similar to personal computer and UNIX operating systems. The integrated file system integrates all information that is stored on the system and allows users and applications to access specific segments of storage that are organized as file, directory, library, and logical unit objects.

The file server allows clients to store and access objects, such as files and programs, that are located on the system. The file server interfaces with the integrated file system and allows clients to use their own interface to interact with files, rather than using the integrated file system interfaces and APIs. Depending on the support provided by the client product, the file server can give clients access to all the files on the system or just the files in the Document Library Services File System (QDLS).

The key features of the integrated file system are the following:

- Support for storing information in stream files, which are files that contain long, continuous strings of data. These strings of data might be, for example, the text of a document or the picture elements in a picture. Documents that are stored in system folders are stream files. Other examples of stream files are PC files and the files in UNIX systems. The stream file support is designed for efficient use in client/server applications.
- A hierarchical directory structure that allows objects to be organized like branches of a tree. To access an object, specify the path from the directories to the object.
- A common interface that allows users and applications to access stream files, database files, documents, and other objects that are stored on the system.

For a list of file systems, see the Work with file systems topic collection. For more information about the integrated file system, see the Integrated file system topic collection.

Related concepts

[File server programs](#)

See a list of associated IBM i Access Client Solutions file server programs with descriptions and associated libraries.

Related information

[Work with file systems](#)

[Integrated file system](#)

File server programs

See a list of associated IBM i Access Client Solutions file server programs with descriptions and associated libraries.

The programs listed in the following table are included with the file server.

File server objects

Program name	Library	Object type	Description
QPWFSEVS0	QSYS	*PGM	Server program
QPWFSEVS2	QSYS	*PGM	Server program
QPWFSEVSD	QSYS	*PGM	Daemon program
QPWFSEV	QSYS	*JOB	Job description used for server jobs
QPWFSEV	QSYS	*CLS	Class used for all file server and database server jobs
QPWFSEVSS	QSYS	*PGM	SSL server program

Related concepts

File server

The IBM i file server and its associated programs work with the integrated file system.

Database server

For Data Transfer, ODBC, System i Navigator database, and IBM i Access Client Solutions providers (OLE DB and the .NET Data provider).

The database server allows clients access to Db2 for i functions. This server provides the following.

- Support for remote SQL access
- Access to data through ODBC, ADO, OLE DB, and .NET Data Provider interfaces
- Database functions (such as creating and deleting files and adding and removing file members)
- Retrieval functions for obtaining information about database files that exist on the system (such as SQL catalog functions)

Additionally, you can use Distributed Relational Database Architecture™ (DRDA) with the database server and with SQL packages. DRDA is not supported by OLE DB or the .NET Data Provider.

Choose from the following topics for more information on working with DRDA. Also, see the Distributed database programming topic collection for additional information about DRDA.

Related information

[Distributed database programming](#)

Database server programs

See a list of IBM i Access Client Solutions database server programs with descriptions and associated libraries.

Program name	Library	Description
QZDASOINIT	QSYS	Server program
QZDASON2	QSYS	Sockets setup program
QZDASRVSD	QSYS	Daemon program
QZDASSINIT	QSYS	SSL server program

Note: The QZDANDB and QZDACRTP *PGM objects along with the *SRVPGM object QZDASRV are used by the database server.

SQL packages

IBM i Access ODBC and the IBM DASH OLE DB Provider support SQL packages.

SQL packages bind SQL statements in an application program to a relational database. They are used to enhance the performance of applications that use dynamic SQL support by allowing the application to reuse information about the SQL requests.

The database server is an application program that uses dynamic SQL requests. It supports the use of packages for frequently used SQL statements so that certain binding information can be reused.

For more information, select from the topics below.

SQL package names

IBM i Access ODBC and OLE DB packages are named differently depending on the database that is accessed.

The database server is sometimes used as a gateway to other relational databases that use DRDA . The database server automatically creates one or more SQL packages on the target relational database. The package names are generated according to the attributes currently used by the database server.

Package name if not a Db2 for i relational database

The package is created in a collection called QSQL400 on the application server, when it is not a Db2 for i relational database (RDB). When it is not a IBM i application server, the package name is QZD **abcde**, in which **abcde** corresponds to specific parser options being used.

The following table shows the options for the package name.

Package name field options

Field	Field description	Options
a	Date format	<ul style="list-style-type: none">• ISO, JIS• USA• EUR• JUL
b	Time format	<ul style="list-style-type: none">• JIS• USA• EUR, ISO
c	Commitment control/ decimal delimiter	<ul style="list-style-type: none">• *CS/period• *CS/comma• *CHG/period• *CHG/comma• *RR/period• *RR/comma
d	String delimiter	<ul style="list-style-type: none">• apostrophe• quote

Field	Field description	Options
e	Maximum number of statements allowed for package	<ul style="list-style-type: none"> • 0 - 64 • 1 - 256 • 2 - 512 • 3 - 1024

Package names if the relational database is Db2 for i

When it is a IBM i application server, the package name is QZDA **abcdef**, in which **abcdef** corresponds to specific parser options being used.

When it is a IBM i RDB, the package is usually created in the QGPL library which most database access clients can customize.

Package name field options

Field	Field description	Options
a	Date format	<ul style="list-style-type: none"> • ISO, JIS • USA • EUR • JUL • MDY • DMY • YMD
b	Time format and naming convention	<ul style="list-style-type: none"> • ISO, JIS and SQL naming • USA and SQL naming • EUR and SQL naming • HMS and SQL naming • ISO, JIS and system naming • USA and system naming • EUR and system naming • HMS and system naming
c	Commit level and decimal point	<ul style="list-style-type: none"> • *CS/period • *CS/comma • *ALL/period • *ALL/comma • *CHG/period • *CHG/comma • *NONE/period • *NONE/comma
d	String delimiter	<ul style="list-style-type: none"> • apostrophe • quote

Field	Field description	Options
e	Number of sections in package	<ul style="list-style-type: none"> • 0 - 64 • 1 - 256 • 2 - 512 • 3 - 1024
f	Date and Time separation	<ul style="list-style-type: none"> • The high order bits of the character: • '1100'b - One of the ISO formats for da • '1101'b - Comma as date separation • '1110'b - Period as date separation • '1111'b - Colon as date separation • The low order bits of the character: • '0001'b - An ISO format of time • '0010'b - Comma as time separator • '0011'b - Period as time separator • '0100'b - Slash as time separator • '0101'b - Dash as time separator • '0110'b - Blank as time separator

Cleanup SQL packages

When using IBM i Access ODBC with DRDA, it is recommended that you periodically use the DLTSQPKG command.

The packages used for DRDA functions are created automatically on your system as needed so you might want to periodically remove these packages. To delete the packages, use the Delete SQL Package (DLTSQPKG) command.

Delete the packages only if they are not used often. The package is created again if needed, but performance noticeably decreases when a package is created a second time.

Statement naming conventions

Identify enforced naming conventions for IBM i database servers.

The following table provides a summary of the naming conventions enforced by the database server.

Statement naming conventions

Statement	Dynamic SQL	Use an extended dynamic SQL package
Local	Statement name must adhere to IBM i naming convention, although the format of STMTxxxx is suggested Cursor name must adhere to IBM i naming conventions	Statement name must adhere to IBM i naming convention, although the format of STMTxxxx is suggested Cursor name must adhere to IBM i naming conventions
DRDA	Statement name must be in the format of STMTxxxx Cursor name must be in the format: CRSRyyyy for non-scrollable cursors or SCRSRyyyy for scrollable cursors where yyyy is the same as xxxx.	Statement name must be in the format of Sxxxx Cursor name must be in the format of Cyy for non-scrollable cursors where yy is the same as xxxx and yy is between 1 and 15.

Notes:

1. The naming convention for statement names is not enforced on the local system, so a client application can share prepared statements with a IBM i application using the QSQPRCED system API.
2. The server appends a blank to the beginning of any statement name in the format of STMTxxxx. A host application must then append a leading blank to share statements with client applications that use the format STMTxxxx. The server does not append a leading blank if the statement name is not in the format of STMTxxxx.

Rules and restrictions when using DRDA

There are Distributed Relational Database Architecture (DRDA) limitations when using IBM i Access Client Solutions database servers.

DRDA is an architecture that supports access between relational databases. For more information on the DRDA architecture, see the Distributed database programming topic collection.

The following table shows the functions that have limitations when you are connected to a remote system from the database server using DRDA.

DRDA functional limits

Function	Limitation
Create package Clear package Delete package Describe parameter markers	Unsupported functions
Prepare	Enhanced prepare option not available when using DRDA.
Extended dynamic package support	<ul style="list-style-type: none"> • When DRDA is used, statement names must be in the format of 'STMTxxxx', where xxxx is the section number. • When DRDA is used, cursor names must be in the format of 'CRSRxxxx' or 'SCRSRxxxx', where xxxx is the section number.
Commit hold	Only valid if a IBM i connection.

Function	Limitation
Commit level *NONE	Not supported
Commit level *CHANGE	Only supported if the target RDB is a IBM i target. All other RDBs require a *CS or *ALL commit level.

Related information

[Distributed database programming](#)

Data queue server

Provides access to IBM i data queues.

A data queue is an object that is used by IBM i application programs for communications. Applications can use data queues to pass data between jobs. Multiple IBM i jobs can send or receive data from a single data queue.

IBM i Access Client Solutions provides APIs that allow PC applications to work with IBM i data queues with the same ease as IBM i applications. This extends IBM i application communications to include processes running on a remote PC.

The programs listed in the following table are included with this server.

Data queue server program provided for use with sockets support

Program name	Library	Description
QZHQSSRV	QSYS	Server program
QZHQSRVD	QSYS	Daemon program

Network print server

Provides remote print support and additional print management when using IBM i Access Client Solutions functions.

The network print server allows enhanced client control over print resources. This print server provides the following capabilities to each client by requesting print serving:

Spooled file

Create, seek, open, read, write, close, hold, release, delete, move, send, call exit program, change attributes, retrieve message, answer message, retrieve attributes, and list

Writer job

Start, end, and list

Printer device

Retrieve attributes and list

Output queue

Hold, release, purge, list, and retrieve attributes

Library

List

Printer file

Retrieve attributes, change attributes, and list

Network print server

Change attributes and retrieve attributes

The programs listed in the following table are included with this server.

Network print server

Program name	Library	Description
QNPSERVS	QSYS	Server program
QNPSERVD	QSYS	Daemon program

Central server

Provides services such as license management and other IBM i Access Client Solutions client management functions.

The central server provides the following services for clients:

- License management

The initial request from either Data Transfer or PC5250 reserves a license for that IBM i Access Client Solutions user. The server remains active until the release delay timeout expires. The license will be held until it is released or the server job is ended. To see which licenses are reserved, use System i Navigator to view the system's properties.

- Retrieve conversion map

The central server retrieves conversion maps for clients who need them. These conversion maps are usually used for ASCII to EBCDIC conversions and for EBCDIC to ASCII conversions. Coded character set identifiers (CCSID) must be supplied. The client can request a map by giving the correct source CCSID, the target CCSID, and a table of code points to be converted. The server then returns the correct mapping for the client to use.

The programs listed in the following table are included with this server.

Central server programs

Program name	Library	Description
QZCSRVS	QSYS	Server program
QZCSRVD	QSYS	Daemon program

Remote command and distributed program call server

Allows PC applications to issue commands and call programs on IBM i and return the results to the client.

The remote command and distributed program call server support allows users and applications to issue CL commands and to call programs. The remote command support allows the user to run multiple commands in the same job. It also offers a better security check for IBM i users with limited capabilities (LMTCPB =*YES, in their user profile).

The distributed program call support allows applications to call IBM i programs and pass parameters (input and output). After the program runs, the output parameter values return to the client application. This process allows applications to access IBM i resources easily without concerns about the communications and conversions that must take place.

The programs listed in the following table are included with this server.

Remote command and distributed program call server programs

Program name	Library	Description
QZRCSRVS	QSYS	Server program
QZRCSRVD	QSYS	Daemon program

Signon server

Provides password management functions for IBM i host servers with sockets support.

The Signon server provides security for clients. This security function prevents access to the system by users with expired passwords, validates user profile passwords and returns user profile security information for use with password caching and System i Navigator Application Administration.

The programs listed in the following table are included with this server.

Signon server programs

Program name	Library	Description
QZSOSIGN	QSYS	Server program
QZSOSGND	QSYS	Daemon program

Server Port Mapper

Provides the current server port number to an IBM i Access Client Solutions client requesting a connection.

The port mapper provides a way for the client to find the port for a particular service (server). The port mapper finds the ports in the TCP/IP Service Table.

The program listed in the following table is included with this server.

Program name	Library	Description
QZSOSMAPD	QSYS	Server port mapper program

Extended Dynamic Remote SQL server (QXDAEDRSQL)

Supports remote IBM i SQL access and other database functions.

The QXDAEDRSQL server allows clients access to Db2 for i functions. This server provides the following.

- Support for remote SQL access
- Access to data through the XDA interface
- Database functions (such as creating and deleting files and adding and removing file members)

The programs listed in the following table are included with this server.

QXDAEDRSQL server programs

Program name	Library	Description
QXDARECVR	QSYS	Server program
QXDALISTEN	QSYS	Daemon program

Note: The QXDAEVT and QXDIAISP *SRVPGM objects are used by the QXDAEDRSQL server.

DRDA/DDM server

Allows access to Db2 for i functions.

The DRDA/DDM server allows clients access to the Db2 for i functions, including record level access when using the OLE DB provider and Toolbox JDBC drivers.

This server provides:

- Support for remote SQL access
- Support for record level access
- Support for remote journal

For more information about DRDA, see Distributed database programming.

For more information about DDM, see Distributed data management.

The programs listed in the following table are included with this server.

DRDA/DDM server programs

Program name	Library	Description
QRWTSRVR	QSYS	Server program
QRWTLSTN	QSYS	Listener program

Related information

[Distributed database programming](#)

[DDM overview](#)

Use IBM i host servers

Describes the client/server communication process, and how to manage it. Additionally, this topic lists relevant IBM i values and subsystems, and describes how to identify, display and manage server jobs on the system.

The servers shipped with the base operating system do not typically require any changes to existing configurations to work correctly. They are set up and configured when you install the IBM i server. You may want to change the way the system manages the server jobs to meet your needs, solve problems, improve system performance, or simply view the jobs on the system. To make such changes and meet processing requirements, you must know which objects affect which pieces of the system and how to change those objects. To really understand how to manage your system, refer to *Work management* before you continue with this topic.

Related information

[Work management](#)

Establish client/server communications

Identify the process for starting and ending communication between IBM i Access Client Solutions clients and host servers.

This topic also includes each server's port numbers, and a description of server daemons and their role in communication.

Client/Server communication is established in the following steps:

1. To initiate a server job that uses sockets communications support, the client system connects to a particular server's port number.
2. A server daemon must be started (with the STRHOSTSVR command) to listen for and accept the client's connection request. Upon accepting the connection request, the server daemon issues an internal request to attach the client's connection to a server job.
3. This server job may be a prestarted job or, if prestart jobs are not used, a batch job that is submitted when the client connection request is processed. The server job handles any further communications with the client. The initial data exchange includes a request that identifies authentication tokens that are associated with the client user. A user profile and password, or a Kerberos ticket, are examples of these tokens.
4. Once the authentication tokens are validated, the server job switches to use the IBM i user profile associated with those tokens, and changes the job by using many of the attributes defined for the user profile, such as accounting code and output queue.

Server to client communications

IBM i Access Client Solutions uses TCP/IP to communicate with the system servers. The optimized servers use IBM i sockets support to communicate with clients. The IBM i sockets support is compatible

with Berkeley Software Distributions 4.3 sockets over TCP/IP. Sockets support is provided with the 5770-TC1 product that is installed on the system.

See the TCP/IP Configuration and Reference manual for more information about communications.

For more information, see:

Related information

TCP/IP setup

Host Servers port numbers

Each type of server has its own server daemon, which listens on a port for incoming IBM i Access Client Solutions client connection requests.

There are exceptions to this. For instance, the transfer function over sockets uses the database server daemon; the network drive server uses the file server daemon; and the virtual print server uses the network print server daemon. In addition, the server mapper daemon also listens on a specified port, and allows a client to obtain the current port number for a specified server.

Each of the server daemons listen on the port number that is provided in the service table for the specified service name. For example, the network print server daemon, with the initial configuration that is provided, listens on port number 8474, which is associated with service name 'as-netprt.' The server mapper daemon listens on the well-known port. The well-known server mapper port number is 449. The well-known port number is reserved for the exclusive use of the Host Servers. Therefore, the entry for the 'as-svrmap' service name should not be removed from the service table.

The port numbers for each server daemon are not fixed; the service table can be modified by using different port numbers if your installation requires such changes. You can change where the port number is retrieved from the System i Navigator system properties connection tab. However, the service name must remain the same as that shown in following tables. Otherwise, the server daemons cannot establish a socket to accept incoming requests for client connection.

If a new service table entry is added to identify a different port number for a service, any pre-existing service table entries for that service name should be removed. Removing these entries eliminates the duplication of the service name in the table and eliminates the possibility of unpredictable results when the server daemon starts.

Port numbers for host servers and server mapper

View port numbers for IBM i Access Client Solutions supported host servers.

The following table shows the initial service table entries provided for the optimized servers and server mapper that use sockets over TCP communication.

Service name	Description	Port number
as-central	Central server	8470
as-database	Database server	8471
as-dtaq	Data queue server	8472
as-file	File server	8473
as-netprt	Network print server	8474
as-rmtcmd	Remote command and program call server	8475
as-signon	Signon server	8476
as-svrmap	Server mapper	449
drda	DRDA/DDM	446
ddm	DRDA/DDM IPsec	447

Each host server type has a server daemon. There is a single server mapper daemon for the system. The client PC application uses the port number to connect to the host server daemon. The server daemon accepts the incoming connection request and routes it to the server job for processing.

Use the CL command finder to see the parameters for the **STRHOSTSVR** command values that are listed below:

Server type

***ALL**

Starts all host server daemons and the server mapper daemon.

***CENTRAL**

Starts the central server daemon in QSYSWRK subsystem. The daemon job is QZSCSRVSD, and the associated server prestart job is QZSCSRVS.

***DATABASE**

Starts the database server daemon in the QSERVER subsystem. The daemon job is QZDASRVSD, and the associated server prestart jobs are QZDASOINIT, QZDASSINIT, and QTFPJTCP. QTFPJTCP runs in the QSERVER subsystem.

***DTAQ**

Starts the data queue server daemon in QSYSWRK subsystem. The daemon job is QZHQSRVD, and the associated server prestart job is QZHQSSRV.

***FILE**

Starts the file server daemon in QSERVER subsystem. The daemon job is QPWFSERVSD, and the associated server prestart jobs are QPWFSERVSO, QPWFSERVSS, and QPWFSERVS2.

***NETPRT**

Starts the network print server daemon in QSYSWRK subsystem. The daemon job is QNPSEVRD, and the associated server prestart jobs are QNPSEVRS and QIWVPPJT. QIWVPPJT runs in the QSYSWRK subsystem.

***RMTCMD**

Starts the remote command and the distributed program call server daemon in QSYSWRK subsystem. The daemon job is QZRCSRVD, and the associated server prestart job is QZRCSRVS.

***SIGNON**

Starts the signon server daemon in QSYSWRK subsystem. The daemon job is QZSOSGND and the associated server prestart job QZSOSIGN.

***SVRMAP**

Starts the server mapper daemon in QSYSWRK subsystem. The daemon job is QZSOSMAPD.

Note: If the daemon job runs in the QSYSWRK directory, the associated server prestart jobs will run in the QUSRWRK directory by default. Additionally, database server prestart jobs will run in QUSRWRK subsystem by default.

Required protocol

(This optional parameter specifies the communication protocols that are required to be active for the host server daemons to start.)

***ANY**

The TCP/IP communication protocol must be active at the time the STRHOSTSVR command is issued. If TCP/IP is not active, diagnostic message PWS3008 and escape message PWS300D are issued and the host server daemons are not started.

***NONE**

No communication protocols need to be active at the time the STRHOSTSVR command is issued for the host server daemons to start. No messages will be issued for protocols which are inactive.

***TCP**

The TCP/IP communication protocol must be active at the time the STRHOSTSVR command is issued. If TCP/IP is not active, diagnostic message PWS3008 and escape message PWS300D are issued and the host server daemons are not started.

Related information

[CL command finder](#)

Server daemons

The server daemon allows IBM i Access Client Solutions client applications to use sockets communications.

The server daemon is a batch job associated with a particular server type. There is only one server daemon for each of the different server types (such as database, network print, and signon). Each server type has a one-to-many relationship between its server daemon and the actual server jobs; one server daemon potentially has many associated server jobs.

The server daemon allows client applications to start communications with a host server that is using sockets communications support. The server daemon does this by handling and routing incoming connection requests. Once the client establishes communications with the server job, there is no further association between the client and the server daemon for the duration of that server job.

Subsystems must be active to use server or file server jobs. When shipped, all server jobs are configured to run in the QUSRWRK subsystem, but you can change the subsystem in which they run. File server jobs and the database host server daemon job (QZDASRVSD) run in the QSERVER subsystem.

The Start Host Server command starts server daemon jobs. The server daemons must be active for client applications to establish a connection with a host server that is using sockets communications support.

If you are starting the database daemon or the file server daemon, the QSERVER subsystem must be active. If you start any of the other server daemons, the QSYSWRK subsystem must be active. To use the prestart jobs for the server daemons that run in the QSYSWRK subsystem, QUSRWRK must be active.

Server Mapper Daemon

The server mapper daemon is a batch job that runs in the QSYSWRK subsystem. It provides a method for client applications to determine the port number associated with a particular server.

This job listens on a well-known port for a connection request from a client. The well-known port number for TCP/IP is 449. The client sends the service name to the server mapper. The server mapper obtains the port number for the specified service name from the service table. The server mapper returns this port number to the client, ends the connection, and returns to listen for another connection request. The client uses the port number returned from the server mapper daemon to connect to the specified server daemon.

The server mapper daemon starts with the STRHOSTSVR command and ends with the ENHOSTSVR command.

Example: STRHOSTSVR

Find examples of using the STRHOSTSVR command when using the IBM i Access Client Solutions product.

Example 1: Starting all host server daemons

```
STRHOSTSVR SERVER(*ALL)
```

This command starts all the server daemons and the server mapper daemon, as long as at least one communication protocol is active.

Example 2: To start specific server daemons

```
STRHOSTSVR SERVER(*CENTRAL *SVRMAP) RQDPCL(*NONE)
```

This command starts the central server daemon and the server mapper daemon, even if no communication protocols are active.

Example 3: Specification of one required protocol

```
STRHOSTSVR SERVER(*ALL) RQDPCL(*TCP)
```

This command starts all the host server daemons and the server mapper daemon, as long as TCP/IP is active.

End host servers

To end IBM i Access Client Solutions host servers, use the **ENDHOSTSVR** CL command.

This command ends the host server daemons and the server mapper daemon. If a server daemon ends while servers of that type are connected to client applications, the server jobs remain active until communication with the client application ends, unless the optional **ENDACTCNN** parameter is specified. Subsequent connection requests from the client application to that server fail until the server daemon starts again.

If the server mapper daemon ends, any existing client connections to server jobs are unaffected. Subsequent requests from a client application to connect to the server mapper fail until the server mapper starts again.

The **ENDACTCNN** parameter may be specified in order to end active connections to the *DATABASE and *FILE servers. This will cause the server jobs that are servicing these connections to end. The active connections can only be ended if the corresponding daemon job is also being ended. If the *DATABASE keyword is specified, the QZDASOINIT and QZDASSINIT jobs with active connections will be ended. If the *FILE keyword is specified, the QPWFSERVSO and QPWFSERVSS jobs with active connections will be ended.

Note: If you use the **ENDHOSTSVR** command to end a particular daemon that is not active, you get a diagnostic message. Use **ENDHOSTSVR SERVER(*ALL)** if you want to end all active daemons. You do not see a diagnostic message with the *ALL value.

ENDHOSTSVR command values:

Server type

***ALL**

Ends the server daemons and the server mapper daemon if active. If used, the system allows no other special values.

***CENTRAL**

Ends the central server daemon in QSYSWRK subsystem.

***DATABASE**

Ends the database server daemon in QSERVER subsystem.

***DTAQ**

Ends the data queue server daemon in QSYSWRK subsystem.

***FILE**

Ends the file server daemon in QSERVER subsystem.

***NETPRT**

Ends the network print server daemon in QSYSWRK subsystem.

***RMTCMD**

Ends the remote command and distributed program call server daemon in QSYSWRK subsystem.

***SIGNON**

Ends the signon server daemon in QSYSWRK subsystem.

***SVRMAP**

Ends the server mapper daemon in QSYSWRK subsystem.

End active connections

(This optional parameter specifies whether the active connections for the specified servers will be ended.)

Single Values:

***NONE**

No active connections will be ended.

Other Values:

***DATABASE**

The active connections being serviced by the QZDASOINIT and QZDASSINIT server jobs will be ended. The server jobs that are servicing these connections will also be ended.

***FILE**

The active connections being serviced by the QPWFSERVSO and QPWFSERVSS server jobs will be ended. The server jobs servicing these connections will also be ended.

Here are some **ENDHOSTSVR** examples.

Example: ENDHOSTSVR

Find examples of using the ENDHOSTSVR command.

Example 1: Ending all host server daemons

```
ENDHOSTSVR SERVER(*ALL)
```

This command ends all the server daemons and the server mapper daemon.

Example 2: To end specific server daemons

```
ENDHOSTSVR SERVER(*CENTRAL *SVRMAP)
```

End the central server daemon and the server mapper daemon.

Example 3: Ending specific server daemons and active connections

```
ENDHOSTSVR SERVER(*CENTRAL *DATABASE) ENDACTCNN(*DATABASE)
```

This command ends the central server daemon in the QSYSWRK subsystem and the database server daemon in the QSERVER subsystem. Additionally, the active connections to the *DATABASE server, and the QZDASOINIT and QZDASSINIT server jobs that are servicing these connections will end.

IBM i Subsystems

System-supplied IBM i subsystems are used to control jobs and functions.

A subsystem description defines how, where, and how much work enters a subsystem, and which resources the subsystem uses to do the work.

Autostart jobs perform one-time initialization or do repetitive work that is associated with a particular subsystem. The autostart jobs associated with a particular subsystem are automatically started each time the subsystem is started.

Related concepts

Identify and display IBM i server jobs

There are different ways to identify and display server jobs.

Subsystems used for server Jobs

The server jobs are configured to run in different subsystems, depending on their function.

The following are the subsystems used for the server jobs.

QSYSWRK

All the daemon jobs (with the exception of the file server daemon job and the database server daemon job) run in this subsystem. The file server and database server daemon jobs run in the QSERVER subsystem.

QUSRWRK

This subsystem is where the server jobs run for these servers:

- Network Print

- Remote command and program call
- Central
- Data Queue
- Signon
- Database

QSERVER

The file server daemon job, its associated prestart server jobs, and the database server daemon job run in this subsystem.

If this subsystem is not active, requests to establish a connection to the file server or the database server will fail.

Automatically starting subsystems

The QSYSWRK subsystem starts automatically when you IPL, regardless of the value specified for the controlling subsystem.

If you use the default startup program provided with the system, the QSERVER and QUSRWRK subsystems start automatically when you IPL. The system startup program is defined in the QSTRUPPGM system value, and the default value is QSTRUP QSYS.

If you want to change the system startup, you can change the QSTRUPPGM system value to call your own program. You can use the shipped program QSTRUP in QSYS as a base for the start-up program that you create.

Note: If you use the database server or file server and you made changes to the system startup, you must ensure that the startup program starts the QSERVER subsystem.

TCP/IP is automatically started by the system without requiring a change to the system startup program. The host servers are automatically started when TCP/IP is started. The IPL attribute, STRTCP, can force the system to not automatically start TCP/IP at IPL. It is recommended to leave this value at the shipped setting of *YES, (start TCP/IP) but the option is available if necessary.

Use of autostart jobs

Autostart jobs are associated with IBM i host servers.

The QSERVER subsystem has an autostart job defined for the file server and database server jobs. If this job is not running, the servers cannot start. The subsystem will not end when the job disappears. If a problem occurs with this job, you may want to end and restart the QSERVER subsystem.

The QSYSWRK subsystem has an autostart job defined for all the optimized servers. This job monitors for events sent when a **STRTCP** command has been issued. This way, the server daemon jobs can dynamically determine when TCP/IP has become active. The daemon jobs then begin to listen on the appropriate ports. If the autostart job is not active, and TCP/IP is started while the host servers are active, the following sequence of commands must be issued to start using TCP/IP:

1. ENHOSTSVR *ALL
2. STRHOSTSVR *ALL

The autostart job is named QZBSEVTM. If the job is not active, it can be started by issuing the following command:

```
QSYS/SBMJOB CMD(QSYS/CALL PGM(QSYS/QZBSEVTM)) JOB(QZBSEVTM) JOBD(QSYS/QZBSEJBD)
PRTDEV(*USRPRF) OUTQ(*USRPRF) USER(QUSER) PRTTXT(*SYSVAL) SYSLIBL(*SYSVAL)
CURLIB(*CRTDFT) INLLIBL(*JOB) SRTSEQ (*SYSVAL) LANGID(*SYSVAL) CNTRYID(*SYSVAL)
CCSID(*SYSVAL)
```

Note: Only one instance of program QZBSEVTM can be running at any one time.

Use of prestart jobs

A prestart job is a batch job that starts running before a program on a remote system initiates communications with the IBM i host server.

Prestart jobs use prestart job entries in the subsystem description to determine which program, class, and storage pool to use when the jobs are started. Within a prestart job entry, you must specify attributes for the subsystem to use to create and to manage a pool of prestart jobs.

Prestart jobs increase performance when you initiate a connection to a server. Prestart job entries are defined within a subsystem. Prestart jobs become active when that subsystem is started, or they can be controlled with the Start Prestart Job (STRPJ) and End Prestart Job (ENDPJ) commands.

System information that pertains to prestart jobs (such as DSPACTPJ) uses the term 'program start request' exclusively to indicate requests made to start prestart jobs, even though the information may pertain to a prestart job that was started as a result of a sockets connection request.

Notes:

- In general, prestart jobs can be reused after they have been returned to the pool. The number of times the prestart job is reused is determined by the value specified for the maximum number of uses (MAXUSE) value of the ADDPJE or CHGPJE CL commands. While it is always best for the connecting client code to clean up the resources that it used while connected to the prestart job, most servers do enough automatic cleanup to safely allow reuse of the prestart job. An exception is the remote command and distributed program call server, which ships with a MAXUSE value of 1. For this server, the resources that are used by one user of the prestart job are not guaranteed to be cleaned up before the prestart job is ended. To prevent these resources from being inadvertently accessed by subsequent users, do not change the MAXUSE value for the remote command and distributed program call server.
- By default, some of the server jobs run in QUSRWRK or QSERVER. Using System i Navigator, you can configure some or all of these servers to run in a subsystem of your choice.

1. **IBM Navigator for i > Network > Servers > IBM i Access.**

2. Right-click the server that you want to configure subsystems for and select **Properties**.

3. Configure the server using the Subsystems page.

If you move jobs from the default subsystem, you must:

1. Create your own subsystem description.

2. Add your own prestart job entries using the ADDPJE command. Set the STRJOBS parameter to *YES.

If you do not do this, your jobs will run in the default subsystem.

All of the host servers that are supported by the sockets communications interface support prestart jobs.

These servers are:

- Network print server

- Remote command and distributed program call server

- Central server

- Database server

- Secure database server

- File server

- Secure file server

- Data queue server

- Signon server (unique to servers using sockets communications support)

The following lists provide each of the prestart job entry attributes and provide the initial values that are configured for the host servers using sockets communications support.

Subsystem description

The subsystem that contains the prestart job entries.

Host server	Value
Network Print	QUSRWRK
Remote command and program call	QUSRWRK
Central	QUSRWRK
Database	QUSRWRK
Secure Database	QUSRWRK
File	QSERVER
Secure File	QSERVER
Data Queue	QUSRWRK
Signon	QUSRWRK

Program library/name

The program that is called when the prestart job is started.

Host server	Value
Network Print	QSYS/QNPSEVS
Remote command and program call	QSYS/QZRCRVS
Central	QSYS/QZSCSRVS
Database	QSYS/QZDASOINIT
Secure Database	QSYS/QZDASSINIT
File	QSYS/QPWFSERVSO
Secure File	QSYS/QPWFSERVSS
Data Queue	QSYS/QZHQSSRV
Signon	QSYS/QZSOSIGN

User profile

The user profile that the job runs under. This is what the job shows as the user profile. When a request to start a server is received from a client, the prestart job function switches to the user profile that is received in that request.

Host server	Value
Network Print	QUSER
Remote command and program call	QUSER
Central	QUSER
Database	QUSER
Secure Database	QUSER
File	QUSER
Secure File	QUSER
Data Queue	QUSER
Signon	QUSER

Job name

The name of the job when it is started.

Host server	Value
Network Print	*PGM
Remote command and program call	*PGM
Central	*PGM
Database	*PGM
Secure Database	*PGM
File	*PGM
Secure File	*PGM
Data Queue	*PGM
Signon	*PGM

Job description

The job description used for the prestart job. Note that if *USRPRF is specified, the job description for the profile that this job runs under will be used. This means QUSER's job description will be used. Some attributes from the requesting user's job description are also used; for example, print device and output queue are swapped from the requesting user's job description.

Host server	Value
Network Print	QSYS/QZBSJOB
Remote command and program call	QSYS/QZBSJOB
Central	QSYS/QZBSJOB
Database	QGPL/QDFTSVR
Secure Database	QGPL/QDFTSVR
File	QGPL/QDFTSVR
Secure File	QGPL/QDFTSVR
Data Queue	QSYS/QZBSJOB
Signon	QSYS/QZBSJOB

Start jobs

Indicates whether prestart jobs are to automatically start when the subsystem is started. These prestart job entries are shipped with a start jobs value of *YES to ensure that the server jobs are available. The STRHOSTSVR command starts each prestart job as part of its processing.

Host server	Value
Network Print	*YES
Remote command and program call	*YES
Central	*YES
Database	*YES
Secure Database	*YES

Host server	Value
File	*YES
Secure File	*YES
Data Queue	*YES
Signon	*YES

Initial number of jobs

The number of jobs that are started when the subsystem starts. This value is adjustable to suit your particular environment and needs.

Host server	Value
Network Print	1
Remote command and program call	1
Central	1
Database	1
Secure Database	1
File	1
Secure File	1
Data Queue	1
Signon	1

Threshold

The minimum number of available prestart jobs for a prestart job entry. When this threshold is reached, additional prestart jobs automatically start. Threshold maintains a certain number of jobs in the pool.

Host server	Value
Network Print	1
Remote command and program call	1
Central	1
Database	1
Secure Database	1
File	1
Secure File	1
Data Queue	1
Signon	1

Additional number of jobs

The number of additional prestart jobs that are started when the threshold is reached.

Host server	Value
Network Print	2
Remote command and program call	2

Host server	Value
Central	2
Database	2
Secure Database	2
File	2
Secure File	2
Data Queue	2
Signon	2

Maximum number of jobs

The maximum number of prestart jobs that can be active for this entry.

Host server	Value
Network Print	*NOMAX
Remote command and program call	*NOMAX
Central	*NOMAX
Database	*NOMAX
Secure Database	*NOMAX
File	*NOMAX
Secure File	*NOMAX
Data Queue	*NOMAX
Signon	*NOMAX

Maximum number of uses

The maximum number of uses of the job. A value of 200 indicates that the prestart job will end after 200 requests to start the server have been processed.

Host server	Value
Network Print	200
Remote command and program call	1
Central	200
Database	200
Secure Database	200
File	*NOMAX
Secure File	*NOMAX
Data Queue	200
Signon	200

Wait for job

This causes a client connection request to wait for an available server job if the maximum number of jobs has been reached.

Host server	Value
Network Print	*YES
Remote command and program call	*YES
Central	*YES
Database	*YES
Secure Database	*YES
File	*YES
Secure File	*YES
Data Queue	*YES
Signon	*YES

Pool identifier

The subsystem pool identifier in which this prestart job runs.

Host server	Value
Network print	1
Remote command and program call	1
Central	1
Database	1
Secure database	1
File	1
Secure file	1
Data queue	1
Signon	1

Class

The name and library of the class the prestart job runs under.

Host server	Value
Network Print	QGPL/QCASERV
Remote command and program call	QGPL/QCASERV
Central	QGPL/QCASERV
Database	QSYS/QPWFSEVER
Secure Database	QSYS/QPWFSEVER
File	QSYS/QPWFSEVER
Secure File	QSYS/QPWFSEVER
Data Queue	QGPL/QCASERV
Signon	QGPL/QCASERV

When the start jobs value for the prestart job entry has been set to *YES and the remaining values are at their initial settings, the following actions take place for each prestart job entry:

you need to change the available number of prestart jobs. If the information indicates that program start requests are waiting for an available prestart job, you can change prestart jobs with the Change Prestart Job Entry (CHGPJE) command.

If the program start requests are not acted on quickly, you can do any combination of the following:

- Increase the threshold
- Increase the parameter value for the initial number of jobs (INLJOBS)
- Increase the parameter value for the additional number of jobs (ADLJOBS)

The key is to ensure that an available prestart job exists for every request.

Remove prestart job entries

If you decide that you do not want the servers to use the prestart job function, you must do the following:

1. End the prestarted jobs with the End Prestart Job (ENDPJ) command.

Prestarted jobs ended with the ENDPJ command are started the next time the subsystem is started if start jobs *YES is specified in the prestart job entry or when the STRHOSTSVR command is issued for the specified server type. If you only end the prestart job and don't take the next step, any requests to start the particular server will fail.

2. Remove the prestart job entries in the subsystem description with the Remove Prestart Job Entry (RMVPJE) command.

The prestart job entries that are removed with the RMVPJE command are permanently removed from the subsystem description. Once the entry is removed, new requests for the server will succeed.

Use routing entries

When a daemon job is routed to a subsystem, the job is using the routing entries in the subsystem description. The routing entries for the host server daemon jobs are added to the subsystem description when the STRHOSTSVR command is issued. These jobs are started under the QUSER user profile. For daemon jobs that are submitted to the QSYSWRK subsystem, the QSYSNOMAX job queue is used. For daemon jobs that are submitted to the QSERVER subsystem, the QPWFSEVER job queue is used.

The characteristics of the server jobs are taken from their prestart job entry. If prestart jobs are not used for the servers, then the server jobs start with the characteristics of their corresponding daemon jobs.

The following information provides the initial configuration in the IBM-supplied subsystems for each of the server daemon jobs.

Network print server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QNPSERVD
Job name	QNPSERVD
Class	QGPL/QCASERVR
Sequence number	2538

Remote command and program call server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZRCSRVD
Job name	QZRCSRVD
Class	QGPL/QCASERV
Sequence number	2539

Central server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZSCSRVD
Job name	QZSCSRVD
Class	QGPL/QCASERV
Sequence number	2536

Database server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSERVER
Job queue	QPWFSEVER
User	QUSER
Route data	QZDASRVSD
Job name	QZDASRVSD
Class	QSYS/QPWFSEVER
Sequence number	600

File server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSERVER
Job queue	QPWFSEVER
User	QUSER
Route data	QPWFSEVRSD
Job name	QPWFSEVRSD
Class	QSYS/QPWFSEVER
Sequence number	200

Data queue server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZHQSRVD
Job name	QZHQSRVD
Class	QGPL/QCASERVR
Sequence number	2537

Signon server daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZSOSGND
Job name	QZSOSGND
Class	QGPL/QCASERVR
Sequence number	2540

Server Mapper daemon

Attribute Name	Attribute Value
Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZSOSMAPD
Job name	QZSOSMAPD
Class	QGPL/QCASERVR
Sequence number	2541

IBM i system values

Learn about the system values that are important in client/server environments.

A system value contains control information that operates certain parts of the system. A user can change the system values to define the work environment. Examples of system values are system date and library list.

There are many system values. The following values are of particular interest in a client/server environment.

QAUDCTL

Audit control. This system value contains the on and off switches for object and user level auditing. Changes that are made to this system value take effect immediately.

QAUDENDACN

Audit journal error action. This system value specifies the action the system takes if errors occur when an audit journal entry is being sent by the operating system security audit journal. Changes that are made to this system value take effect immediately.

QAUDFRCLVL

Force audit journal. This system value specifies the number of audit journal entries that can be written to the security auditing journal before the journal entry data is forced to auxiliary storage. Changes that are made to this system value take effect immediately.

QAUDLVL

Security auditing level. Changes made to this system value take effect immediately for all jobs running on the system.

QAUTOVRT

Determines whether the system should automatically create virtual devices. This is used with display station pass-through and Telnet sessions.

QCCSID

The coded character set identifier, which identifies:

- A specific set of encoding scheme identifiers
- Character set identifiers
- Code page identifiers
- Additional coding-related information that uniquely identifies the coded graphic character representation needed by the system

This value is based on the language that is installed on the system. It determines whether data must be converted to a different format before being presented to the user. The default value is 65535, which means this data is not converted.

QCTLSBSD

The controlling subsystem description

QDSPSGNINF

Determines whether the sign-on information display shows after sign-on by using the 5250 emulation functions (workstation function, PC5250).

QLANGID

The default language identifier for the system. It determines the default CCSID for a user's job if the job CCSID is 65535. The clients and servers use this default job CCSID value to determine the correct conversion for data that is exchanged between the client and the server.

QLMTSECOFR

Controls whether a user with all-object (*ALLOBJ) or service (*SERVICE) special authority can use any device. If this value is set to 1, all users with *ALLOBJ or *SERVICE special authorities must have specific *CHANGE authority to use the device.

This affects virtual devices for 5250 emulation. The shipped value for this is 1. If you want authorized users to sign-on to PCs, you must either give them specific authority to the device and controller that the PC uses or change this value to 0.

QMAXSIGN

Controls the number of consecutive incorrect sign-on attempts by local and remote users. Once the QMAXSIGN value is reached, the system determines the action with the QMAXSGNACN system value.

If the QMAXSGNACN value is 1 (vary off device), the QMAXSIGN value does not affect a user who enters an incorrect password on the PC when they are starting the connection.

This is a potential security exposure for PC users. The QMAXSGNACN should be set to either 2 or 3.

QMAXSGNACN

Determines what the system does when the maximum number of sign-on attempts is reached at any device. You can specify 1 (vary off device), 2 (disable the user profile) or 3 (vary off device and disable the user profile). The shipped value is 3.

QPWDEXPITV

The number of days for which a password is valid. Changes that are made to this system value take effect immediately.

QPWDLMTAJC

Limits the use of adjacent numbers in a password. Changes that are made to this system value take effect the next time a password is changed.

QPWDLMTCHR

Limits the use of certain characters in a password. Changes that are made to this system value take effect the next time a password is changed.

QPWDLMTREP

Limits the use of repeating characters in a password. Changes that are made to this system value take effect the next time a password is changed.

QPWDLVL

Determines the level of password support for the system, which includes the password length that is supported, the type of encryption used for passwords, and whether IBM i NetServer passwords for the Windows clients are removed from the system. Changes that are made to this system value take effect on the next IPL.

QPWDMAXLEN

The maximum number of characters in a password. Changes that are made to this system value take effect the next time a password is changed.

QPWDMINLEN

The minimum number of characters in a password. Changes that are made to this system value take effect the next time a password is changed.

QPWDPOSDIF

Controls the position of characters in a new password. Changes that are made to this system value take effect the next time a password is changed.

QPWDRQDDGT

Requires a number in a new password. Changes that are made to this system value take effect the next time a password is changed.

QPWDRQDDIF

Controls whether the password must be different than previous passwords.

QPWDVLDPGM

Password validation program name and library that are supplied by the computer system. Both an object name and library name can be specified. Changes that are made to this system value take effect the next time a password is changed.

QRMTSIGN

Specifies how the system handles remote sign-on requests. A TELNET session is actually a remote sign-on request. This value determines several actions, as follows:

- '*FRCSIGNON!': All remote sign-on sessions are required to go through normal sign-on processing.
- '*SAMEPRF!': For 5250 display station pass-through or workstation function, when the source and target user profile names are the same, the sign-on may be bypassed for remote sign-on attempts. When using TELNET, the sign-on may be bypassed.
- '*VERIFY!': After verifying that the user has access to the system, the system allows the user to bypass the sign-on.
- '*REJECT!': Allows no remote sign-on for 5250 display station pass-through or workstation function. TELNET uses a value of *FRCSIGNON when *REJECT is specified and displays the sign-on display. If you want to reject all TELNET requests to the system, end the TELNET servers.
- '*program library*!': The user can specify a program and library (or *LIBL) to decide which remote sessions are allowed and which user profiles can be automatically signed on from which locations. This option is only used for passthrough. TELNET uses a value of *FRCSIGNON when this program is specified.

This value also specifies a program name to run that determines which remote sessions are to be allowed.

The shipped value is *FRCSIGNON. If you want users to be able to use the bypass sign-on function of the 5250 emulator, change this value to *VERIFY.

QSECURITY

System security level. Changes that are made to this system value take effect at the next IPL.

- 20 means that the system requires a password to sign-on.
- 30 means that the system requires password security at sign-on and object security at each access. You must have authority to access all system resources.
- 40 means that the system requires password security at sign-on and object security at each access. Programs that try to access objects through unsupported interfaces fail.
- 50 means that the system requires password security at sign-on, and users must have authority to access objects and system resources. The security and integrity of the QTEMP library and user domain objects are enforced. Programs that try to access objects through interfaces that are not supported or that try to pass unsupported parameter values to supported interfaces will fail.

QSTRUPPGM

The program that runs when the controlling subsystem starts or when the system starts. This program performs set up functions such as starting subsystems.

QSYSLIBL

The system part of the library list. This part of the library list is searched before any other part. Some client functions use this list to search for objects.

Identify and display IBM i server jobs

There are different ways to identify and display server jobs.

Identifying a particular job is a prerequisite to investigating problems and determining performance implications.

You can use an emulator or a character-based interface. You can also use the IBM Navigator for i interface to identify your server jobs if you prefer using a graphical user interface (GUI). You might find it easier to relate a job to a certain personal computer or an individual client function using the GUI interface. Both the character-based and the GUI method allow you to identify and work with your server jobs.

Related concepts

IBM i Subsystems

System-supplied IBM i subsystems are used to control jobs and functions.

Use IBM Navigator for i to identify server jobs

You can display and work with server jobs.

Follow these steps to use the IBM Navigator for i interface to identify your server jobs.

1. Open **Network** by clicking the **plus sign (+)**.
2. Open **Servers** by clicking the **plus sign (+)**.
3. Select the type of servers for which you want to see jobs (For example, TCP/IP or IBM i Access Client Solutions).
4. When the servers show in the right pane, right-click on the server for which you want to see jobs and click **Server Jobs**. Another window opens, showing the server jobs with the user, job type, job status, time entered system and date entered system for that server.

IBM i job names

IBM i job names follow a specific naming convention.

The job name consists of three parts:

- The simple job name
- The user ID

- The job number (ascending order)

The server jobs follow several conventions:

- Job name
 - For jobs that are not prestarted, the server job name is the name of the server program.
 - Prestarted jobs use the name that is defined in the prestart job entry.
 - Jobs that are started by the servers use the job description name or a given name if they are batch jobs (the file server does this).
- The user ID
 - Is always QUSER, regardless of whether prestart jobs are used.
 - The job log shows which users have used the job.
- Work management creates the job number.

Display using WRKACTJOB

Use the IBM i **WRKACTJOB** command to display server jobs.

The **WRKACTJOB** command shows all active jobs, as well as the server daemons and the server mapper daemon.

The following figures show a sample status with the **WRKACTJOB** command. Only jobs related to the servers are shown in the figures. You must press **(F14)** to see the available prestart jobs.

The following types of jobs are shown in the figures:

- **(1)** - Server mapper daemon
- **(2)** - Server daemons
- **(3)** - Prestarted server jobs

```

-----
                          Work with Active Jobs                               LP11UT11
                                                                07/28/09 15:13:08
CPU %:   3.1   Elapsed time: 21:38:40   Active jobs: 77
Type options, press Enter.
  2=Change  3=Hold  4=End   5=Work with  6=Release  7=Display message
  8=Work with spooled files 13=Disconnect ...

Opt  Subsystem/Job  User      Type  CPU %  Function      Status
----
(1)  QSYSWRK        QSYS      SBS   .0     .         DEQW
(1)  QZSOSMAPD     QUSER     BCH   .0     .         SELW
(2)  QZSOSGND       QUSER     BCH   .0     .         SELW
(2)  QZSCSRVSD     QUSER     BCH   .0     .         SELW
(2)  QZRCSRVD      QUSER     BCH   .0     .         SELW
(2)  QZHQSRVD     QUSER     BCH   .0     .         SELW
(2)  QNPSERVD     QUSER     BCH   .0     .         SELW
.
(3)  QUSRWRK        QSYS      SBS   .0     .         DEQW
(3)  QZSOSIGN      QUSER     PJ    .0     .         PSRW
(3)  QZSCSRVS     QUSER     PJ    .0     .         PSRW
(3)  QZRCSRVS     QUSER     PJ    .0     .         PSRW
(3)  QZHQSSRV     QUSER     PJ    .0     .         PSRW
(3)  QNPSERVS     QUSER     PJ    .0     .         PSRW
(3)  QZDASOINIT   QUSER     PJ    .0     .         PSRW
.
More...
-----

```

```

-----
                          Work with Active Jobs                               LP11UT11
                                                                07/28/09 15:13:08
CPU %:   3.1   Elapsed time: 21:38:40   Active jobs: 77
Type options, press Enter.
-----

```

Opt	Subsystem/Job	User	Type	CPU %	Function	Status
	QSERVER	QSYS	SBS	.0		DEQW
	QSERVER	QPGMR	ASJ	.1		EVTW
(2)	QPWFSEVSD	QUSER	BCH	.0		SELW
	QZDASRVSD	QUSER	BCH	.0		SELW
(3)	QPWFSEVSO	QUSER	PJ	.0		PSRW
	QPWFSEVSO	QUSER	PJ	.0		PSRW
						More...

The following types of jobs are shown:

ASJ

The autostart job for the subsystem

PJ

The prestarted server jobs

SBS

The subsystem monitor jobs

BCH

The server daemon and the server mapper daemon jobs

Display using the history log

Find IBM i server jobs by using the history log.

Each time a client user successfully connects to a server job, that job is swapped to run under the profile of that client user.

To determine which job is associated with a particular client user, you can display the history log with the **DSPLOG** command. Look for the messages starting with:

- CPIAD0B (for signon server messages)
- CPIAD09 (for messages relating to all other servers)

Display server job for a user

Use IBM Navigator for i or the **WRKOBJLCK** command.

Follow these steps to display the server jobs for a particular user, using IBM Navigator for i:

1. Access the following URL from a Web browser where *systemName* is your IBM i name: `http://systemName:2001`.
2. Click on **Users and Groups**, then **Users**.
3. Right-click on the user that you want to see server jobs for.
4. Select **User Objects**, then click on **Jobs**. You see a window displaying all the server jobs for that user.

You can also use the **WRKOBJLCK** command to find all of the server jobs for a particular user. To use the command, specify the user profile as the object name, and *USRPRF as the object type.

Use server exit programs

Write and register exit programs when using IBM i host servers.

Exit programs allow system administrators to control which activities a client user is allowed for each of the specific servers. All of the servers support user-written exit programs. This topic describes how the exit programs can be used, and how to configure them. It also provides sample programs that can help control access to server functions.

Note: By using the code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 105.

Register exit programs

Identify IBM i exit programs to call.

Work with the registration facility

For the servers to know which exit program, if any, to call, you must register your exit program. You can register the exit program using the IBM i registration facility.

In addition to registering an exit program, it is necessary to restart the prestart jobs for a particular server. Without this step, the exit program is not called until, through attrition, new server jobs start. For the file server exit program to be invoked, the QSERVER subsystem must be restarted.

To register an exit program with the registration facility, use the Work with Registration Information (WRKREGINF) command.

```
+-----+
|                                     |
|                               Work with Registration Info (WRKREGINF) |
|                                     |
| Type choices, press Enter.         |
|                                     |
| Exit point . . . . . *REGISTERED  |
| Exit point format . . . . . *ALL   | Name, generic*, *ALL |
| Output . . . . . *                | *, *PRINT          |
|                                     |
+-----+
```

Press Enter to view the registered exit points.

```
+-----+
|                                     |
|                               Work with Registration Information |
|                                     |
| Type options, press Enter.         |
|   5=Display exit point   8=Work with exit programs |
|                                     |
|                                     |
|                                     |
| Opt  Exit Point Format Registered Text |
| -    QIBM_QCA_CHG_COMMAND CHGC0100 *YES Change command exit programs |
| -    QIBM_QCA_RTV_COMMAND RTVC0100 *YES Retrieve command exit progra |
| -    QIBM_QHQ_DTAQ DTAQ0100 *YES Original data queue server |
| -    QIBM_QIMG_TRANSFORMS XFRM0100 *YES |
| -    QIBM_QJO_DLT_JRNRCV DRCV0100 *YES Delete Journal Receiver |
| -    QIBM_QLZP_LICENSE LICM0100 *YES Original License Mgmt Server |
| -    QIBM_QMF_MESSAGE MESS0100 *YES Original Message Server |
| -    QIBM_QMH_REPLY_INQ RPYI0100 *YES Handle reply to inquiry mess |
| 8    QIBM_QNPS_ENTRY ENTR0100 *YES Network Print Server - entry |
| -    QIBM_QNPS_SPLF SPLF0100 *YES Network Print Server - spool |
| -    QIBM_QOE_OV_USR_ADM UADM0100 *YES OfficeVision/400 Administrat |
|                                     |
| Command |
| =====> |
|                                     |
+-----+
```

Choose option 8 to work with the exit programs for the exit point defined for the server you would like to work with.

```
+-----+
|                                     |
|                               Work with Exit Programs |
|                                     |
| Exit point:  QIBM_QNPS_ENTRY          Format:  ENTR0100 |
|                                     |
| Type options, press Enter.         |
|   1=Add   4=Remove   5=Display   10=Replace |
|                                     |
|                                     |
| Opt  Exit Program Exit Program Library |
| 1_   Number          |
|                                     |
| (No exit programs found) |
|                                     |
+-----+
```

Use option 1 to add an exit program to an exit point.

Notes:

- If an exit program is already defined, you must remove it before you can change the name of the program.
- Even though the registration facility can support multiple user exits for a specific exit point and format name, the servers always retrieve exit program 1.
- You must end and restart the prestart jobs for the change to go into effect.

```

+-----+
|                                     Add exit program (ADDEXITPGM)
|
| Type choices, press Enter.
|
| Exit point . . . . . > QIBM_QNPS_ENTRY
| Exit point format . . . . . > ENTR0100      Name
| Program number . . . . . > 1                1-2147483647, *LOW, *HIGH
| Program . . . . . MYPGM                      Name
| Library . . . . . MYLIB                      Name, *CURLIB
| THREADSAFE . . . . . *UNKNOWN              *UNKNOWN, *NO, *YES
| Multithreaded job action . . . . . *SYSVAL  *SYSVAL, *RUN, *MSG,
| Text 'description' . . . . . *BLANK
+-----+

```

Enter your program name and library for the program at this exit point.

The same program is usable for multiple exit points. The program can use the data that is sent as input to determine how to handle different types of requests.

The following table provides the exit point and format names for each of the specific IBM i servers.

<i>Table 2. Server exit point to format mapping</i>		
Server	Format name(s)	Application name
QIBM_QPWFS_FILE_SERV (File Server)	PWFS0100	*FILESRV
QIBM_QZDA_INIT (Database server initiation)	ZDAI0100	*SQL
QIBM_QZDA_NDB1 (Database server-native database requests)	ZDAQ0100 ZDAQ0200	*NDB
QIBM_QZDA_ROI1 (Database server retrieve object information requests)	ZDAR0100 ZDAR0200	*RTVOBJINF
QIBM_QZDA_SQL1 (Database server SQL requests)	ZDAQ0100	*SQLSRV
QIBM_QZDA_SQL2 (Database server SQL requests)	ZDAQ0200	*SQLSRV
QIBM_QZHQ_DATA_QUEUE (Data queue server)	ZHQ00100	*DATAQSRV
QIBM_QNPS_ENTRY (Network print server)	ENTR0100	QNPSEVR
QIBM_QNPS_SPLF (Network print server)	SPLF0100	QNPSEVR
QIBM_QZSC_LM (Central server license management requests)	ZSCL0100	*CNTRLSRV
QIBM_QZSC_NLS (Central server NLS requests)	ZSCN0100	*CNTRLSRV
QIBM_QZSC_SM (Central server client system management)	ZSCS0100	*CNTRLSRV
QIBM_QZRC_RMT (Remote command and distributed program call server)	CZRC0100	*RMTSRV

<i>Table 2. Server exit point to format mapping (continued)</i>		
Server	Format name(s)	Application name
QIBM_QZSO_SIGNONSRV (Signon server)	ZSOY0100	*SIGNON

Write exit programs

This topic identifies considerations when specifying IBM i exit programs.

When you specify an exit program the servers pass the following two parameters to the exit program before running your request:

- A 1-byte return code value
- A structure containing information about your request (This structure is different for each of the exit points.)

These two parameters allow the exit program to determine whether your request is possible. If the exit program sets the return code to X'F1', the server allows the request. If the return code is set to X'F0' the server rejects the request. If values other than X'F1' or X'F0' are set, the results will vary depending upon which server is being accessed.

For multiple servers and exit points, the same program is usable. The program can determine which server is being called and which function is being used by looking at the data in the second parameter structure.

See the Exit program parameters topic collection which documents the structures of the second parameter that is sent to the exit programs. You can use this information to write your own exit programs.

Related concepts

[Exit program parameters](#)
Identify IBM i exit points.

Exit program parameters

Identify IBM i exit points.

These topics provide the data structure for the second parameter of the exit point formats for each of the host servers.

Related concepts

[Write exit programs](#)
This topic identifies considerations when specifying IBM i exit programs.

File server

Identify IBM i exit points for file serving.

The file server has one defined exit point, QIBM_QPWFS_FILE_SERV, which supports two formats: PWFS0100 and PWFS0200. Only one exit program is called by the server for a function. If a program is registered for both formats, only the program that is registered for format PWFS0200 is called.

The QIBM_QPWFS_FILE_SERV exit point is defined to run an exit program for the following types of file server requests:

- Change file attributes
- Create stream file or create directory
- Delete file or delete directory
- List file attributes
- Move
- Open stream file
- Rename
- Allocate conversation

- Copy (format PWFS0200 only)

Notes:

- For the file server, the exit program name is resolved when the QSERVER subsystem is activated. If you change the program name, you must end and restart the subsystem for the change to take effect.
- For file server requests that provide the file name to the exit program, the user must have a minimum of *RX authority to each directory in the path name preceding the object. If the user does not have the required authority, the request fails.
- If the file server exit program swaps to another user and does not swap back to the original user, the file server session continues to operate with the user that originally connected to the session. This is because the host file server and IBM i NetServer get credential information for the user who did the initial connection to the session and uses this credential information when doing client requests. With the host file server and IBM i NetServer using the credential information, any swapping of the user profile in the file server exit program is not used by the file server for file system operations.

Exit point QIBM_QPWFS_FILE_SERV format PWFS0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the file server, the value is *FILESRV.
20	14	BINARY(4)	Requested function	The function being performed: <ul style="list-style-type: none"> • X'0000' - Change file attributes request • X'0001' - Create stream file or directory request • X'0002' - Delete file or delete directory request • X'0003' - List file attributes request • X'0004' - Move request • X'0005' - Open stream file request • X'0006' - Rename request • X'0007' - Allocate conversation request
24	18	CHAR(8)	Format name	The user exit format name being used. For QIBM_QPWFS_FILE_SERV, the format name is PWFS0100.
32	20	CHAR(4)	File access	If the requested function has a value of X'0005' (open), this field contains the following structure: <ul style="list-style-type: none"> • Read access, CHAR(1) X'F1' - Yes X'F0' - No • Write access, CHAR(1) X'F1' - Yes X'F0' - No • Read/write access, CHAR(1) X'F1' - Yes X'F0' - No • Delete allowed, CHAR(1) X'F1' - Yes X'F0' - No
36	24	BINARY(4)	File name length	The length of the file name (the next field). The length can be a maximum of 16 MB. If the requested function has a value of X'0007' (Allocate conversation request), the file name length is 0.

Offset		Type	Field	Description
Dec	Hex			
40	28	CHAR(*)	File name	<p>The name of the file. The length of this field is specified by the file name length (the previous field). The file name is returned in CCSID 1200.</p> <p>If a requested function has a value of one of the following, the file name is provided and the file name length is set:</p> <ul style="list-style-type: none"> • X'0000' - Change file attributes request • X'0001' - Create stream file or directory request • X'0002' - Delete file or delete directory request • X'0003' - List file attributes request • X'0004' - Move request • X'0005' - Open stream file request • X'0006' - Rename request
<p>Notes:</p> <ul style="list-style-type: none"> • This format is defined by member EPWFSEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC, and QCBLLSRC in library QSYSINC. • The APIs available to convert to and from CCSID 1200 are iconv() and CDCRCVRT. 				

Exit point QIBM_QPWFS_FILE_SERV format PWFS0200

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the file server, the value is *FILESRV.
20	14	BINARY(4)	Requested function	<p>The function being performed:</p> <ul style="list-style-type: none"> • X'0000' - Change file attributes request • X'0001' - Create stream file or directory request • X'0002' - Delete file or delete directory request • X'0003' - List file attributes request • X'0004' - Move request • X'0005' - Open stream file request • X'0006' - Rename request • X'0007' - Allocate conversation request • X'0008' - Copy request
24	18	CHAR(8)	Format name	The user exit format name being used. For QIBM_QPWFS_FILE_SERV, the format name is PWFS0200.
32	20	CHAR(4)	File access	<p>If the requested function has a value of X'0005' (open), this field contains the following structure:</p> <ul style="list-style-type: none"> • Read access, CHAR(1) X'F1' - Yes X'F0' - No • Write access, CHAR(1) X'F1' - Yes X'F0' - No • Read/write access, CHAR(1) X'F1' - Yes X'F0' - No • Delete allowed, CHAR(1) X'F1' - Yes X'F0' - No

Offset		Type	Field	Description
Dec	Hex			
36	24	CHAR(10)	Object type	The type of object being created. If the requested function has a value of X'0001' (Create stream file or directory request), the object type is set to one of the following values. <ul style="list-style-type: none"> • *STMF " - Stream file being created • *DIR " - Directory being created
46	2E	CHAR(6)	Reserved	Unused area.
52	34	BINARY(4)	File name offset	The offset from the start of the format to the name of the object. Zero is set if the requested function has a value of X'0007' (Allocate conversation request).
56	38	BINARY(4)	File name length	The length of the file name. The length can be a maximum of 16 MB. If the requested function has a value of X'0007' (Allocate conversation request), the file name length is 0.
60	3C	BINARY(4)	Target file name offset	The offset from the start of the format to the target name of the object. The offset is set to a nonzero value if the requested function has a value of X'0004' (Move), x'0006' (Rename), or X'0008' (Copy request).
64	40	BINARY(4)	Target file name length	The length of the target file name. The length is set to zero when the target name offset is set to zero.
*	*	CHAR(*)	File name	The name of the file. The length of this field is specified by the file name length. The file name is returned in CCSID 1200. The file name is set if the file name offset field has a value other than zero.
*	*	CHAR(*)	Target file name	The name of the target file. The length of this field is specified by the target file name length field. The file name is returned in CCSID 1200. The target file name is set if the target file name offset field has a value other than zero.
Notes:				
<ul style="list-style-type: none"> • This format is defined by member EPWFSEP in files H, QRPGRS, QRPGLSRC, QLBSRC, and QCBLESRC in library QSYSINC. • The APIs available to convert to and from CCSID 1200 are iconv() and CDRCVRT. • The length of this format, including file names, can be a maximum size of 16 MB. 				

Database server

Identify exit points for IBM i database serving.

The database server has five different exit points defined:

1. QIBM_QZDA_INIT
 - Called at server initiation
2. QIBM_QZDA_NDB1
 - Called for native database requests
3. QIBM_QZDA_SQL1
 - Called for SQL requests
4. QIBM_QZDA_SQL2
 - Called for SQL requests
5. QIBM_QZDA_ROI1

- Called for retrieving object information requests and SQL catalog functions

The exit points for native database and retrieving object information have two formats defined depending on the type of function requested.

The QIBM_QZDA_INIT exit point is defined to run an exit program at server initiation. If a program is defined for this exit point, it is called each time the database server is initiated.

Exit point QIBM_QZDA_INIT format ZDAI0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For this exit point, the value is *SQL.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZDA_INIT the format name is ZDAI0100.
28	1C	BINARY(4)	Requested function	The function being performed The only valid value for this exit point is 0.
32	20	CHAR(63)	Interface type	The type of interface connecting to the server job.
95	5F	CHAR(127)	Interface name	The name of the interface connecting to the server job.
222	DE	CHAR(63)	Interface level	The level of the interface connecting to the server job.
<p>Note: This format is defined by member EZDAEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.</p>				

The QIBM_QZDA_NDB1 exit point is defined to run an exit program for native database requests for the database server. Two formats are defined for this exit point. Format ZDAD0100 is used for the following functions:

- Create source physical file
- Create database file, based on existing file
- Add, clear, delete database file member
- Override database file
- Delete database file override
- Delete file

Format ZDAD0200 is used when a request is received to add libraries to the library list.

Exit point QIBM_QZDA_NDB1 format ZDAD0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For this exit point, the value is *NDB.

Offset		Type	Field	Description
Dec	Hex			
20	14	CHAR(8)	Format name	The user exit format name being used For the following functions, the format name is ZDAD0100.
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: <ul style="list-style-type: none"> • X'00001800' - Create source physical file • X'00001801' - Create database file • X'00001802' - Add database file member • X'00001803' - Clear database file member • X'00001804' - Delete database file member • X'00001805' - Override database file • X'00001806' - Delete database file override • X'00001807' - Create save file • X'00001808' - Clear save file • X'00001809' - Delete file
32	20	CHAR(128)	File name	Name of the file used for the requested function
160	A0	CHAR(10)	Library name	Name of the library that contains the file
170	AA	CHAR(10)	Member name	Name of the member to be added, cleared, or deleted
180	B4	CHAR(10)	Authority	Authority to the created file
190	BE	CHAR(128)	Based on file name	Name of the file to use when creating a file based on an existing file
318	13E	CHAR(10)	Based on library name	Name of the library containing the based on file
328	148	CHAR(10)	Override file name	Name of the file to be overridden
338	152	CHAR(10)	Override library name	Name of the library that contains the file to be overridden
348	15C	CHAR(10)	Override member name	Name of the member to be overridden
<p>Note: This format is defined by member EZDAEP in files H, QRPGRS, QRPGLSRC, QLBSRC and QCBLESRC in library QSYSINC.</p>				

Exit point QIBM_QZDA_NDB1 format ZDAD0200

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For this exit point, the value is *NDB.
20	14	CHAR(8)	Format name	The user exit format name being used. For the add to library list function, the format name is ZDAD0200.
28	1C	BINARY(4)	Requested function	The function being performed X'0000180C' - Add library list
32	20	BINARY(4)	Number of libraries	The number of libraries (the next field)
36	24	CHAR(10)	Library name	The library names for each library
Note: This format is defined by member EZDAEP in files H, QRPGRSRC, QRPGLSRC, QLBSRC and QCBLLSRC in library QSYSINC.				

The QIBM_QZDA_SQL1 exit point is defined to run an exit point for certain SQL requests that are received for the database server. Only one format is defined for this exit point. The following are the functions that cause the exit program to be called:

- Prepare
- Open
- Execute
- Connect
- Create package
- Clear package
- Delete package
- Stream fetch
- Execute immediate
- Prepare and describe
- Prepare and execute or prepare and open
- Open and fetch
- Execute or open
- Return package information

Exit point QIBM_QZDA_SQL1 format ZDAQ0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For this exit point, the value is *SQLSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZDA_SQL1, the format name is ZDAQ0100.

Offset		Type	Field	Description
Dec	Hex			
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: <ul style="list-style-type: none"> • X'00001800' - Prepare • X'00001803' - Prepare and describe • X'00001804' - Open/Describe • X'00001805' - Execute • X'00001806' - Execute immediate • X'00001809' - Connect • X'0000180C' - Stream fetch • X'0000180D' - Prepare and execute • X'0000180E' - Open and fetch • X'0000180F' - Create package • X'00001810' - Clear package • X'00001811' - Delete package • X'00001812' - Execute or open • X'00001815' - Return package information
32	20	CHAR(18)	Statement name	Name of the statement used for the prepare or execute functions
50	32	CHAR(18)	Cursor name	Name of the cursor used for the open function. If the actual cursor name is greater than 18 bytes, the following special value is passed, indicating that the cursor name is obtained from the <i>Extended cursor name</i> field: <ul style="list-style-type: none"> • *EXTDCRSR
68	44	CHAR(2)	Prepare option	Option used for the prepare function
70	46	CHAR(2)	Open attributes	Option used for the open function
72	48	CHAR(10)	Extended dynamic package name	Name of the extended dynamic SQL package
82	52	CHAR(10)	Package library name	Name of the library for extended dynamic SQL package.
92	5C	BINARY(2)	DRDA indicator	<ul style="list-style-type: none"> • 0 - Connected to local RDB • 1 - Connected to remote RDB
94	5E	CHAR(1)	Isolation level	<ul style="list-style-type: none"> • 'A' - Commit *ALL • 'C' - Commit *CHANGE • 'N' - Commit *NONE • 'L' - Commit *RR (repeatable read) • 'S' - Commit *CS (cursor stability)

Offset		Type	Field	Description
Dec	Hex			
95	5F	CHAR(512)	First 512 bytes of the SQL statement text	First 512 bytes of the SQL statement
607	25F	CHAR(258)	Extended cursor name	The extended cursor name
Note: This format is defined by member EZDAEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.				

The QIBM_QZDA_SQL2 exit point is defined to run an exit point for certain SQL requests that are received for the database server. The QIBM_QZDA_SQL2 exit point takes precedence over the QIBM_QZDA_SQL1 exit point. If a program is registered for the QIBM_QZDA_SQL2 exit point, it will be called and a program for the QIBM_QZDA_SQL1 exit point will not be called. The following are the functions that cause the exit program to be called:

- Prepare
- Open
- Execute
- Connect
- Create package
- Clear package
- Delete package
- Stream fetch
- Execute immediate
- Prepare and describe
- Prepare and execute or prepare and open
- Open and fetch
- Execute or open
- Return package information

Table A-6. Exit point QIBM_QZDA_SQL2 format ZDAQ0200

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For this exit point, the value is *SQLSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZDA_SQL2, the format name is ZDAQ0200.

Offset		Type	Field	Description
Dec	Hex			
28	1C	BINARY(4)	Requested function	<p>The function being performed</p> <p>This field contains one of the following:</p> <ul style="list-style-type: none"> • X'00001800' - Prepare • X'00001803' - Prepare and describe • X'00001804' - Open/Describe • X'00001805' - Execute • X'00001806' - Execute immediate • X'00001809' - Connect • X'0000180C' - Stream fetch • X'0000180D' - Prepare and execute • X'0000180E' - Open and fetch • X'0000180F' - Create package • X'00001810' - Clear package • X'00001811' - Delete package • X'00001812' - Execute or open • X'00001815' - Return package information
32	20	CHAR(18)	Statement name	Name of the statement used for the prepare or execute functions
50	32	CHAR(18)	Cursor name	<p>Name of the cursor used for the open function. If the actual cursor name is greater than 18 bytes, the following special value is passed, indicating that the cursor name is obtained from the <i>Extended cursor name</i> field:</p> <ul style="list-style-type: none"> • *EXTDCRSR
68	44	CHAR(2)	Prepare option	Option used for the prepare function
70	46	CHAR(2)	Open attributes	Option used for the open function
72	48	CHAR(10)	Extended dynamic package name	Name of the extended dynamic SQL package
82	52	CHAR(10)	Package library name	Name of the library for extended dynamic SQL package.
92	5C	BINARY(2)	DRDA indicator	<ul style="list-style-type: none"> • 0 - Connected to local RDB • 1 - Connected to remote RDB
94	5E	CHAR(1)	Isolation level	<ul style="list-style-type: none"> • 'A' - Commit *ALL • 'C' - Commit *CHANGE • 'N' - Commit *NONE • 'L' - Commit *RR (repeatable read) • 'S' - Commit *CS (cursor stability)

Offset		Type	Field	Description
Dec	Hex			
95	5F	CHAR(10)	Default SQL schema	Name of the default SQL schema used by the Database Server
105	69	CHAR(3)		Reserved
108	6C	BINARY(4)	Offset to the extended cursor name	The offset int his structure to the extended cursor name
112	70	BINARY(4)	Length of the extended cursor name	The length, in bytes, of the extended cursor name
116	74	CHAR(118)		Reserved
234	EA	BINARY(4)	SQL statement text length	Length of SQL statement text in the field that follows. The length can be a maximum of 2 MB (2,097,152 bytes).
238	EE	CHAR(*)	SQL statement text	Entire SQL statement
*	*	CHAR(*)	Extended cursor name	The extended cursor name

Note: This format is defined by member EZDAEP in files H, QRP GSRC, QRP GLE SRC, QL BLSRC and QCB LLESRC in library QSYSINC.

The QIBM_QZDA_ROI1 exit point is defined to run an exit program for the requests that retrieve information about certain objects for the database server. It is also used for SQL catalog functions.

This exit point has two formats defined. These formats are described below.

Format ZDAR0100 is used for requests to retrieve information for the following objects:

- Library (or collection)
- File (or table)
- Field (or column)
- Index
- Relational database (or RDB)
- SQL package
- SQL package statement
- File member
- Record format
- Special columns

Format ZDAR0200 is used for requests to retrieve information for the following objects:

- Foreign keys
- Primary keys

Exit point QIBM_QZDA_ROI1 format ZDAR0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the database server, the value is *RTVOBJINF.
20	14	CHAR(8)	Format name	The user exit format name being used. For the following functions, the format name is ZDAR0100.
28	1C	BINARY(4)	Requested function	<p>The function being performed</p> <p>This field contains one of the following:</p> <ul style="list-style-type: none"> • X'00001800' - Retrieve library information • X'00001801' - Retrieve relational database information • X'00001802' - Retrieve SQL package information • X'00001803' - Retrieve SQL package statement • X'00001804' - Retrieve file information • X'00001805' - Retrieve file member information • X'00001806' - Retrieve record format information • X'00001807' - Retrieve field information • X'00001808' - Retrieve index information • X'0000180B' - Retrieve special column information
32	20	CHAR(20)	Library name	The library or search pattern used when retrieving information about libraries, packages, package statements, files, members, record formats, fields, indexes, and special columns
52	34	CHAR(36)	Relational database name	The relational database name or search pattern used to retrieve RDB information
88	58	CHAR(20)	Package name	The package name or search pattern used to retrieve package or package statement information
108	6C	CHAR(256)	File name (SQL alias name)	The file name or search pattern used to retrieve file, member, record format, field, index, or special column information
364	16C	CHAR(20)	Member name	The member name or search pattern used to retrieve file member information

Offset		Type	Field	Description
Dec	Hex			
384	180	CHAR(20)	Format name	The format name or search pattern used to retrieve record format information
Note: This format is defined by member EZDAEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.				

Exit point QIBM_QZDA_ROI1 format ZDAR0200

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the database server, the value is *RTVOBJINF.
20	14	CHAR(8)	Format name	The user exit format name being used. For the following functions, the format name is ZDAR0200.
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: <ul style="list-style-type: none"> • X'00001809' - Retrieve foreign key information • X'0000180A' - Retrieve primary key information
32	20	CHAR(10)	Primary key table library name	The name of the library that contains the primary key table used when retrieving primary and foreign key information
42	2A	CHAR(128)	Primary key table name (alias name)	The name of the table that contains the primary key used when retrieving primary or foreign key information
170	AA	CHAR(10)	Foreign key table library name	The name of the library that contains the foreign key table used when retrieving foreign key information
180	64	CHAR(128)	Foreign key table name (alias name)	The name of the table that contains the foreign key used when retrieving foreign key information
Note: This format is defined by member EZDAEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.				

Data queue server

Identify the IBM i exit point for data queue serving.

The data queue server has one exit point defined:

QIBM_QZHQ_DATA_QUEUE format ZHQ00100

The exit point QIBM_QZHQ_DATA_QUEUE is defined to run an exit point program when the following data queue server requests are received:

- Query
- Receive
- Create
- Delete
- Send
- Clear
- Cancel
- Peek

Exit point QIBM_QZHQ_DATA_QUEUE format ZHQ00100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the data queue, server the value is *DATAQSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZHQ_DATA_QUEUE the format name is ZHQ00100.
28	1C	BINARY(4)	Requested function	<p>The function being performed</p> <ul style="list-style-type: none"> • X'0001' - Query the attributes of a data queue • X'0002' - Receive a message from a data queue • X'0003' - Create a data queue • X'0004' - Delete a data queue • X'0005' - Send a message to a data queue • X'0006' - Clear messages from a data queue • X'0007' - Cancel a pending receive request • X'0012' - Receive a message from a data queue without deleting it
32	20	CHAR(10)	Object name	Data queue name
42	2A	CHAR(10)	Library name	Data queue library

Offset		Type	Field	Description
Dec	Hex			
52	34	CHAR(2)	Relational operation	Relational operator for receive-by-key operation on the request X'0000' - No operator 'EQ' - Equal 'NE' - Not equal 'GE' - Greater or equal 'GT' - Greater than 'LE' - Less or equal 'LT' - Less than
54	36	BINARY(4)	Key length	Key length specified on the request
58	3A	CHAR(256)	Key value	Key value specified on the request
Note: This format is defined by member EZHQEP in files H, QRPGRSRC, QRPGLSRC, QLBSRC and QCBLLSRC in library QSYSINC.				

Network print server

Identify IBM i exit points for network print serving.

The network print server has two exit points defined:

1. QIBM_QNPS_ENTRY format ENTR0100
 - Called at server initiation
2. QIBM_QNPS_SPLF format SPLF0100
 - Called to process an existing spooled output file

The QIBM_QNPS_ENTRY exit point is defined to run an exit program when the network print server is started. The exit program can be used to verify access to the server. For more information, see *Printer Device Programming*, SC41-5713-03.

Exit point QIBM_QNPS_ENTRY format ENTR0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the network print server, the value is QNPSERVER.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QNPS_ENTRY the format name is ENTR0100.
28	1C	BINARY(4)	Function identifier	The function being performed For QIBM_QNPS_ENTRY the value is X'0802'.
Note: This format is defined by member ENPSEP in files H, QRPGRSRC, QRPGLSRC, QLBSRC and QCBLLSRC in library QSYSINC.				

The QIBM_QNPS_SPLF exit point is defined to run an exit program after the network print server receives a request to process an existing spooled output file. The program can be used to perform a function on the spooled file, such as fax the file. For more information, see *Printer Device Programming*, SC41-5713-03.

Exit point QIBM_QNPS_SPLF format SPLF0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the network print server the value is QNPSERVER
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QNPS_SPLF the format name is SPLF0100.
28	1C	BINARY(4)	Function identifier	The function being performed For QIBM_QNPS_SPLF, the value is 'X'010D'.
32	20	CHAR(10)	Job name	The name of the job that created the spooled file
42	2A	CHAR(10)	User name	The user profile of the job that created the spooled file
52	34	CHAR(6)	Job number	The number of the job that created the spooled file
58	3A	CHAR(10)	Spooled file name	The name of the spooled file being requested
68	44	BINARY(4)	Spooled file number	The number of the spooled file being requested
72	48	BINARY(4)	Length	Length of the spooled file exit program data
76	4C	CHAR(*)	Spooled file exit program data	Spooled file exit program data consists of additional information used by the exit program that has registered for exit point QIBM_QNPS_SPLF. The client application provides the spooled file exit program data.

Note: This format is defined by member ENPSEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.

Central server

Identify IBM i exit points for the central server.

The central server has three exit points defined:

1. QIBM_QZSC_LM format ZSCL0100
 - Called for license management requests
2. QIBM_QZSC_SM format ZSCS0100

- Called for system management requests
3. QIBM_QZSC_NLS format ZSCN0100

- Called for conversion table requests

The QIBM_QZSC_LM exit point is defined to run an exit program for all license management requests received by the central server.

Exit program QIBM_QZSC_LM format ZSCL0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the central server, the value is *CNTRLSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZSC_LM, the format name is ZSCL0100.
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: <ul style="list-style-type: none"> • X'1001' - Request license • X'1002' - Release license • X'1003' - Retrieve license information
32	20	CHAR(255)	Unique client name	The unique client name is used to identify a specific workstation across a network. The use of a licensed product is assigned to a workstation identified by the unique client name.
287	11F	CHAR(8)	License user handle	License user handle is used to ensure that the license requester and license releaser are the same. This value must be the same as when the license was requested.
295	127	CHAR(7)	Product identification	The identification of the product whose licensed use is requested
302	12E	CHAR(4)	Feature identification	The feature of the product
306	132	CHAR(6)	Release identification	The version, release, and modification level of the product or feature
312	138	BINARY(2)	Type of information	The type of information to be retrieved. The type of information field is only valid for the retrieve license information function This field contains one of the following: <ul style="list-style-type: none"> • X'0000' - Basic license information • X'0001' - Detailed license information

Offset		Type	Field	Description
Dec	Hex			
Note: This format is defined by member EZSCEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.				

The QIBM_QZSC_SM exit point is defined to run an exit program for all client management requests received by the central server.

Exit program QIBM_QZSC_SM format ZSCS0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the central server, the value is *CNTRLSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZSC_SM the format name is ZSCS0100.
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: <ul style="list-style-type: none"> • X'1101' - Set client active • X'1102' - Set client inactive
32	20	CHAR(255)	Unique client name	The client workstation name that is assigned to the licensed product
287	11F	CHAR(255)	Community name	The community name SNMP configuration field is used for authentication.
542	21E	CHAR(1)	Node type	The type of connection <ul style="list-style-type: none"> • 3 - Internet
543	21F	CHAR(255)	Node name	The name of the node For node type 3, the node name will be an Internet address.
Note: This format is defined by member EZSCEP in files H, QRPGRSRC, QRPGLSRC, QLBLSRC and QCBLLSRC in library QSYSINC.				

The QIBM_QZSC_NLS exit point is defined to run an exit program when the central server receives a request to retrieve a conversion map.

Exit program QIBM_QZSC_NLS format ZSCN0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server

Offset		Type	Field	Description
Dec	Hex			
10	A	CHAR(10)	Server identifier	For the central server, the value is *CNTRLSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZSC-NLS, the format name is ZSCN0100.
28	1C	BINARY(4)	Requested function	The function being performed <ul style="list-style-type: none"> • X'1201' - Retrieve conversion map
32	20	BINARY(4)	From coded character set identifier (CCSID)	CCSID for existing data
36	24	BINARY(4)	To coded character set identifier (CCSID)	CCSID into which the data will be converted
40	28	BINARY(2)	Type of conversion	Requested mapping type: <ul style="list-style-type: none"> • X'0001' - Round trip • X'0002' - Substitution mapping • X'0003' - Best-fit mapping
<p>Note: This format is defined by member EZSCEP in files H, QRPGRS, QRPGLSRC, QLBSRC and QCBLLSRC in library QSYSINC.</p>				

Remote command and distributed program call server

Identify the IBM i exit point for the remote command and the distributed program call server

The remote command and distributed program call server has one exit point defined:

QIBM_QZRC_RMT format CZRC0100

The QIBM_QZRC_RMT exit point is defined to call a program for either remote command or distributed program call requests.

The format of the parameter fields differ according to the type of request.

Remote command requests for exit point QIBM_QZRC_RMT format CZRC0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the remote command server, the value is *RMTSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZRC_RMT, the format name is CZRC0100.
28	1C	BINARY(4)	Requested function	The function being performed <ul style="list-style-type: none"> • X'1002' - Remote command

Offset		Type	Field	Description
Dec	Hex			
32	20	BINARY(4)	CCSID of command string	The CCSID of the command string. Valid values are: <ul style="list-style-type: none"> • 0 - Job CCSID • 1200 - UTF-16 • 1208 - UTF - 8
36	24	CHAR(16)	Reserved	Not used for remote command requests
52	34	BINARY(4)	Length of the next field	The length of the following command string
56	38	CHAR (*)	Command string	Command string for remote command requests

Distributed program call requests for exit point QIBM_QZRC_RMT format CZRC0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server
10	A	CHAR(10)	Server identifier	For the distributed program call server, the value is *RMTSRV.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZRC_RMT, the format name is CZRC0100.
28	1C	BINARY(4)	Requested function	The function being performed X'1003' - Distributed program call
32	20	CHAR(10)	Program name	Name of the program being called
42	2A	CHAR(10)	Library name	Library of the specified program
52	34	BINARY(4)	Number of parameters	The total number of parameters for the program call. This does not always indicate the number of parameters that follow.

Offset		Type	Field	Description
Dec	Hex			
56	38	CHAR(*)	Parameter information	<p>Information about the parameters being passed to the specified program. All parameter strings have the following format regardless of the parameter usage type. The last field in the structure is specified for input/output parameter usage types.</p> <ul style="list-style-type: none"> • BINARY(4) - Length of parameter information for this parameter • BINARY(4) - Maximum length of parameter • BINARY(2) - Parameter usage type <ul style="list-style-type: none"> – 1 - Input – 2 - Output – 3 - Input / output • CHAR(*) - Parameter string

Signon server

Identify the IBM i exit point for the signon server.

The signon server has one exit point defined:

QIBM_QZSO_SIGNONSRV format ZSOY0100

The exit point QIBM_QZSO_SIGNONSRV is defined to run an exit point program when the following signon server requests are received:

- Start server request
- Retrieve sign-on information
- Change password
- Generate authentication token
- Generate authentication token on behalf of another user

Exit point QIBM_QZSO_SIGNONSRV format ZSOY0100

Offset		Type	Field	Description
Dec	Hex			
0	0	CHAR(10)	User profile name	The name of the user profile associated with the request
10	A	CHAR(10)	Server identifier	For the signon server, the value is *SIGNON.
20	14	CHAR(8)	Format name	The user exit format name being used. For QIBM_QZSO_SIGNONSRV, the format name is ZSOY0100.

Offset		Type	Field	Description
Dec	Hex			
28	1C	BINARY(4)	Requested function	The function being performed <ul style="list-style-type: none"> • X'7002' - Start server request • X'7004' - Retrieve sign-on information • X'7005' - Change password • X'7007' - Generate authentication token • X'7008' - Generate authentication token on behalf of another user

Examples: Exit programs

These sample IBM i exit programs do not show all possible programming considerations or techniques, but you can review the examples before you begin your own design and coding.

Code example disclaimer

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

All sample code is provided by IBM for illustrative purposes only. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

All programs contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

Examples: Creating exit programs with RPG

Use IBM i exit programs with RPG.

The following example illustrates how to set up a user exit program with RPG*.

Note: By using the code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 105.

```

**
** IBM i SERVERS - SAMPLE USER EXIT PROGRAM
**
** THE FOLLOWING RPG PROGRAM UNCONDITIONALLY
** ACCEPTS ALL REQUESTS. IT CAN BE USED AS A SHELL
** FOR SPECIFIC APPLICATIONS. NOTE: REMOVE THE
** SUBROUTINES AND CASE STATEMENT ENTRIES FOR THE SERVERS
** THAT DO NOT REQUIRE
** SPECIFIC EXIT PROGRAM HANDLING FOR BETTER PERFORMANCE.
**
E*
E* NECESSARY ARRAY DEFINITIONS FOR TRANSFER FUNCTION
E* AND REMOTE SQL
E*
E          TFREQ    4096  1
E          RSREQ    4107  1
I*
I*
IPCSDTA      DS
I              1  10  USERID
I              11  20  APPLID
I*
I* SPECIFIC PARAMETERS FOR VIRTUAL PRINTER
I*
I              21  30  VPFUNC
I              31  40  VPOBJ
I              41  50  VPLIB
I              71  750VPIFN

```

```

I          76 85 VPOUTQ
I          86 95 VPQLIB
I*

```

```

I* SPECIFIC PARAMETERS FOR MESSAGING FUNCTION
I          21 30 MFFUNC
I*

```

```

I* SPECIFIC PARAMETERS FOR TRANSFER FUNCTION
I*
I          21 30 TFFUNC
I          31 40 TFOBJ
I          41 50 TFLIB
I          51 60 TFMBR
I          61 70 TFFMT
I          71 750TFLEN
I          764171 TREQ
I*

```

```

I* SPECIFIC PARAMETERS FOR FILE SERVER
I*
I* NOTE: FSNAME MAY BE UP TO 16MB.
I* FSNLEN WILL CONTAIN THE ACTUAL SIZE OF FSNAME.
I*

```

```

I          B 21 240FSFID
I          25 32 FSFMT
I          33 33 FSREAD
I          34 34 FSWRIT
I          35 35 FSRDWR
I          36 36 FSDLT
I          B 37 400FSNLEN
I          41 296 FSNAME
I*

```

```

I* SPECIFIC PARAMETERS FOR DATA QUEUES
I*
I          21 30 DQFUNC
I          31 40 DQQ
I          41 50 DQLIB
I          70 750DQLEN
I          76 77 DQROP
I          78 820DQKLEN
I          83 338 DQKEY
I*

```

```

I* SPECIFIC PARAMETERS FOR REMOTE SQL
I*
I          21 30 RSFUNC
I          31 40 RSOBJ
I          41 50 RSLIB
I          51 51 RSCMT
I          52 52 RSMODE
I          53 53 RSCID
I          54 71 RSSTN
I          72 75 RRSRV
I          764182 RSREQ
I*

```

```

I* SPECIFIC PARAMETERS FOR NETWORK PRINT SERVER
I*
I          21 28 NPFT
I          B 29 320NPFID

```

```

I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT SPLF0100
I          33 42 NPJOB#
I          43 52 NPUSR#
I          53 58 NPJOB#
I          59 68 NPFILE
I          B 69 720NPFIL#
I          B 73 760NPLEN
I          77 332 NPDATA
I*

```

```

I* Data queue server:
I*
I* QIBM_QZHQ_DATA_QUEUE format ZHQ00100
I*

```

```

I          21 28 DQOFMT
I          B 29 320DQOFID
I          33 42 DQOOBJ
I          43 52 DQOLIB
I          53 54 DQOROP
I          B 55 580DQOLEN
I          59 314 DQOKEY
I*

```

```

I* Specific PARAMETERS FOR CENTRAL SERVER

```

```

I*
I          21 28 CSFMT
I          B 29 320CSFID
I* Central server:
I*
I* QIBM_QZSC_LM format ZSCL0100 for license management calls
I*
I*
I          33 287 CSLCNM
I          288 295 CSLUSR
I          296 302 CSLPID
I          303 306 CSLFID
I          307 312 CSLRID
I          B 313 3140CSLTYP
I*
I* Central server:
I*
I* QIBM_QZSC_LM format ZSCS0100 for system management calls
I*
I*
I          33 287 CSSCNM
I          288 542 CSSCMY
I          543 543 CSSNDE
I          544 798 CSSNNM
I*

```

```

I* Central server:
I*
I* QIBM_QZSC_LM format ZSCN0100 for retrieve conversion map calls
I*
I*
I          21 30 CSNXFM
I          29 320CSNFNC
I          B 33 360CSNFRM
I          B 37 400CSNTO
I          B 41 420CSNCNT
I*
I* SPECIFIC PARAMETERS FOR DATABASE SERVER
I*
I          21 28 DBFMT
I          B 29 320DBFID
I*
I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0100
I          33 160 DBDFIL
I          161 170 DBDLIB
I          171 180 DBDMBR
I          181 190 DBDAUT
I          191 318 DBDBFL
I          319 328 DBDBLB
I          329 338 DBDOFL
I          339 348 DBDOLB
I          349 358 DBDOMB
I*
I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0200
I          B 33 360DBNUM
I          37 46 DBLIB2
I*
I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAQ0100
I          33 50 DBSTMT
I          51 68 DBCRSR
I          69 70 DBOPI
I          71 72 DBATTR
I          73 82 DBPKG
I          83 92 DBPLIB
I          B 93 940DBDRDA
I          95 95 DBCMT
I          96 351 DBTEXT
I* THE FOLLOWING PARAMETERS REPLACE DBTEXT FOR FORMAT ZDAQ0200
I          96 105 DBSQCL
I          B 133 1360DBSQLN
I          137 392 DBSQTX
I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAR0100
I          33 52 DBLIBR
I          53 88 DBRDBN
I          89 108 DBPKGR
I          109 364 DBFILR
I          365 384 DBMBRR
I          385 404 DBFFT

```

```

I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAR0200
I          33  42 DBRPLB
I          43 170 DBRPTB
I          171 180 DBRFLB
I          181 308 DBRFTB
I*
I* Remote command and distributed program call server:
I*
I* QIBM_QZRC_RMT format CZRC0100
I*   RCPGM AND RCLIB ARE NOT USED FOR REMOTE COMMAND CALLS
I*
I          21  28 RCFMT
I          B  29 320RCFID
I          33  42 RCPGM
I          43  52 RCLIB
I          B  53 560RCNUM
I          57 312 RCDATA
I*
I* signon server:
I*
I* QIBM_QZSO_SIGNONSRV format ZSOY0100 for TCP/IP signon server
I*
I          21  28 SOXFMT
I          B  29 320SOFID
I*

```

I*****

```

I*
I          '*VPRT'      '      C          #VPRT
I          '*TRFCL'   '      C          #TRFCL
I          '*FILESRV' '      C          #FILE
I          '*MSGFCL'  '      C          #MSGF
I          '*DQSRV'   '      C          #DQSRV
I          '*RQSRV'   '      C          #RQSRV
I          '*SQL'     '      C          #SQL
I          '*NDB'     '      C          #NDBSV
I          '*SQLSRV'  '      C          #SQLSV
I          '*RTVOBJINF' '      C          #RTVOB
I          '*DATAQSRV' '      C          #DATAQ
I          '*QNPSEVR' '      C          #QNPSV
I          '*CNTRLSRV' '      C          #CNTRL
I          '*RMTSRV'  '      C          #RMTSV
I          '*SIGNON'  '      C          #SIGN
I*

```

```

I*
C*
C* EXIT PROGRAM CALL PARAMETERS
C*
C          *ENTRY    PLIST
C                   PARM          RTNCD  1
C                   PARM          PCSDTA
C*

```

```

C* INITIALIZE RETURN VALUE TO ACCEPT REQUEST

```

```

C*
C          MOVE '1'      RTNCD
C*

```

```

C* COMMON PROCESSING

```

```

C*
C          COMMON LOGIC GOES HERE
C*

```

```

C* PROCESS BASED ON SERVER ID

```

```

C*
C          APPLID    CASEQ#VPRT    VPRT
C          APPLID    CASEQ#TRFCL   TFR
C          APPLID    CASEQ#FILE     FILE
C          APPLID    CASEQ#MSGF     MSG
C          APPLID    CASEQ#DQSRV    DATAQ
C          APPLID    CASEQ#RQSRV    RSQ
C          APPLID    CASEQ#SQL       SQLINT
C          APPLID    CASEQ#NDBSV    NDB
C          APPLID    CASEQ#SQLSV    SQLSRV
C          APPLID    CASEQ#RTVOB    RTVOBJ
C          APPLID    CASEQ#DATAQ    ODATAQ
C          APPLID    CASEQ#QNPSV    NETPRT
C          APPLID    CASEQ#CNTRL    CENTRL
C          APPLID    CASEQ#RMTSV    RMTCMD
C          APPLID    CASEQ#SIGN     SIGNON
C
C          END
C          SETON                      LR
C          RETRN

```

```

C*
C* SUBROUTINES
C*
C*
C* VIRTUAL PRINT
C*
C          VPRT          BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*
C* TRANSFER FUNCTION
C*
C* THE FOLLOWING IS AN EXAMPLE OF SPECIFIC PROCESSING
C* THAT THE EXIT PROGRAM COULD DO FOR TRANSFER FUNCTION.
C*

```

```

C* IN THIS CASE, USERS ARE NOT ALLOWED TO SELECT
C* DATA FROM ANY FILES THAT ARE IN LIBRARY QIWS.
C*

```

```

C          TFR          BEGSR
C          TFFUNC       IFEQ 'SELECT'
C          TFLIB        ANDEQ 'QIWS'
C          MOVE '0'          RTNCD
C          END
C          ENDSR
C*
C*

```

```

C* FILE SERVER
C*

```

```

C          FILE          BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* MESSAGING FUNCTION
C*

```

```

C          MSG          BEGSR
C*          SPECIFIC LOGIC GOFS HERE
C          ENDSR
C*

```

```

C* DATA QUEUES
C*

```

```

C          DATAQ       BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* REMOTE SQL
C*

```

```

C          RSQL          BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* SERVERS
C*

```

```

C* DATABASE INIT
C*

```

```

C          SQLINT       BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* DATABASE NDB (NATIVE DATABASE)
C*

```

```

C          NDB          BEGSR
C*          SFECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* DATABASE SQL
C*

```

```

C          SQLSRV       BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* DATABASE RETRIEVE OBJECT INFORMATION
C*

```

```

C          RTVOBJ       BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* DATA QUEUE SERVER
C*

```

```

C          ODATAQ      BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*
C* NETWORK PRINT
C*
C          NETPRT      BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*
C* CENTRAL SERVER
C*
C*
C* THE FOLLOWING IS AN EXAMPLE OF SPECIFIC PROCESSING
C* THAT THE EXIT PROGRAM COULD DO FOR LICENSE MANAGEMENT.
C*
C* IN THIS CASE, THE USER "USERALL" WILL NOT BE ALLOWED
C* TO EXECUTE ANY FUNCTIONS THAT ARE PROVIDED BY THE
C* CENTRAL SERVER FOR WHICH THIS PROGRAM IS A REGISTERED
C* EXIT PROGRAM - LICENSE INFORMATION, SYSTEM MANAGEMENT
C* OR RETRIVE A CONVERSION MAP.
C*
C          CENTRL      BEGSR
C          USERID      IFEQ 'USERALL'
C                      MOVE '0'          RTNCD
C                      ENDIF
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C*

```

```

C* REMOTE COMMAND AND DISTRIBUTED PROGRAM CALL
C*
C* IN THIS CASE, THE USER "USERALL" WILL NOT BE ALLOWED
C* TO EXECUTE ANY REMOTE COMMANDS OR REMOTE PROGRAM CALLS
C*
C          RMTCMD      BEGSR
C          USERID      IFEQ 'USERALL'
C                      MOVE '0'          RTNCD
C                      ENDIF
C          ENDSR
C*
C* SIGNON SERVER
C*
C          SIGNON      BEGSR
C*          SPECIFIC LOGIC GOES HERE
C          ENDSR
C

```

Examples: Create exit programs with CL commands

You can create IBM i exit programs using CL commands.

The following example illustrates how to set up a user exit program with control language (CL) commands.

Note: Read the Code example disclaimer for important legal information.

```

/*****/
/*                                          */
/* IBM i - SAMPLE USER EXIT PROGRAM          */
/*                                          */
/* THE FOLLOWING CL PROGRAM UNCONDITIONALLY  */
/* ACCEPTS ALL REQUESTS. IT CAN BE USED AS A SHELL FOR DEVELOPING */
/* EXIT PROGRAMS TAILORED FOR YOUR OPERATING ENVIRONMENT.          */
/*                                          */
/*                                          */
/*****/
PGM PARM(&STATUS &REQUEST)

/* * * * * * * * * * * * * * * * * */
/*                                          */
/* PROGRAM CALL PARAMETER DECLARATIONS */
/*                                          */
/* * * * * * * * * * * * * * * * * */

DCL VAR(&STATUS) TYPE(*CHAR) LEN(1) /* Accept/Reject indicator */
DCL VAR(&REQUEST) TYPE(*CHAR) LEN(9999) /* Parameter structure. LEN(9999) is a CL limit.*/

```

```

/*****
/*
/* PARAMETER DECLARES
/*
/*****

/* COMMON DECLARES */
DCL VAR(&USER) TYPE(*CHAR) LEN(10)
/* User ID */
DCL VAR(&APPLIC) TYPE(*CHAR) LEN(10)
/* Server ID */
DCL VAR(&FUNCTN) TYPE(*CHAR) LEN(10) /* Function being performed */

```

```

/* VIRTUAL PRINT DECLARES */
DCL VAR(&VPOBJ) TYPE(*CHAR) LEN(10) /* Object name */
DCL VAR(&VPLIB) TYPE(*CHAR) LEN(10) /* Object library name */
DCL VAR(&VPLEN) TYPE(*DEC) LEN(5 0) /* Length of following fields*/
DCL VAR(&VPOUTQ) TYPE(*CHAR) LEN(10) /* Output queue name */
DCL VAR(&VPQLIB) TYPE(*CHAR) LEN(10) /* Output queue library name */

```

```

/* TRANSFER FUNCTION DECLARES */
DCL VAR(&TFOBJ) TYPE(*CHAR) LEN(10) /* Object name */
DCL VAR(&TFLIB) TYPE(*CHAR) LEN(10) /* Object library name */
DCL VAR(&TFMBR) TYPE(*CHAR) LEN(10) /* Member name */
DCL VAR(&TFMT) TYPE(*CHAR) LEN(10) /* Record format name */
DCL VAR(&TFLEN) TYPE(*DEC) LEN(5 0) /* Length of request */
DCL VAR(&TFREQ) TYPE(*CHAR) LEN(1925) /*Transfer request
statement*/

```

```

/* FILE SERVER DECLARES */
DCL VAR(&FSFID) TYPE(*CHAR) LEN(4) /* Function identifier */
DCL VAR(&FSFMT) TYPE(*CHAR) LEN(8) /* Parameter format */
DCL VAR(&FSREAD) TYPE(*CHAR) LEN(1) /* Open for read */
DCL VAR(&FSWRITE) TYPE(*CHAR) LEN(1) /* Open for write */
DCL VAR(&FSRDWRT) TYPE(*CHAR) LEN(1) /* Open for read/write */
DCL VAR(&FSDLT) TYPE(*CHAR) LEN(1) /* Open for delete */
DCL VAR(&FSLLEN) TYPE(*CHAR) LEN(4) /* fname length */
DCL VAR(&FSNAME) TYPE(*CHAR) LEN(2000) /* Qualified file name */

```

```

/* DATA QUEUE DECLARES */
DCL VAR(&DQQ) TYPE(*CHAR) LEN(10) /* Data queue name */
DCL VAR(&DQLIB) TYPE(*CHAR) LEN(10) /* Data queue library name */
DCL VAR(&DQLEN) TYPE(*DEC) LEN(5 0) /* Total request length */
DCL VAR(&DQROP) TYPE(*CHAR) LEN(2) /* Relational operator */
DCL VAR(&DQKLEN) TYPE(*DEC) LEN(5 0) /* Key length */
DCL VAR(&DQKEY) TYPE(*CHAR) LEN(256) /* Key value */

```

```

/* REMOTE SQL DECLARES */
DCL VAR(&RSOBJ) TYPE(*CHAR) LEN(10) /* Object name */
DCL VAR(&RSLIB) TYPE(*CHAR) LEN(10) /* Object library name */
DCL VAR(&RSCMT) TYPE(*CHAR) LEN(1) /* Commitment control level*/
DCL VAR(&RSMODE) TYPE(*CHAR) LEN(1) /* Block/Update mode indicator*/
DCL VAR(&RSCID) TYPE(*CHAR) LEN(1) /* Cursor ID */
DCL VAR(&RSSTN) TYPE(*CHAR) LEN(18) /* Statement name */
DCL VAR(&RSRSU) TYPE(*CHAR) LEN(4) /* Reserved */
DCL VAR(&RSREQ) TYPE(*CHAR) LEN(1925) /* SQL statement */

```

```

/* NETWORK PRINT SERVER DECLARES */
DCL VAR(&NPFMT) TYPE(*CHAR) LEN(8) /* Format name */
DCL VAR(&NPFID) TYPE(*CHAR) LEN(4) /* Function identifier*/
/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT SPLF0100 */
DCL VAR(&NPJOBN) TYPE(*CHAR) LEN(10) /* Job name */
DCL VAR(&NPUSRN) TYPE(*CHAR) LEN(10) /* User name */
DCL VAR(&NPJOB#) TYPE(*CHAR) LEN(6) /* Job number */
DCL VAR(&NPFILE) TYPE(*CHAR) LEN(10) /* File name */
DCL VAR(&NPFIL#) TYPE(*CHAR) LEN(4) /* File number */
DCL VAR(&NPLEN) TYPE(*CHAR) LEN(4) /* Data Length */
DCL VAR(&NPDATA) TYPE(*CHAR) LEN(2000) /* Data */

DCL VAR(&DBNUM) TYPE(*CHAR) LEN(4) /* Number of libraries */
DCL VAR(&DBLIB2) TYPE(*CHAR) LEN(10) /* Library name */

```

```

/* DATA QUEUE SERVER DECLARES */
DCL VAR(&DQFMT) TYPE(*CHAR) LEN(8) /* Format name */
DCL VAR(&DQFID) TYPE(*CHAR) LEN(4) /* Function IDENTIFIER */
DCL VAR(&DQOOBJ) TYPE(*CHAR) LEN(10) /* Object name */
DCL VAR(&DQOLIB) TYPE(*CHAR) LEN(10) /* Library name */
DCL VAR(&DQOROP) TYPE(*CHAR) LEN(2) /* Relational operator */
DCL VAR(&DQOLEN) TYPE(*CHAR) LEN(4) /* Key length */
DCL VAR(&DQOKEY) TYPE(*CHAR) LEN(256) /* Key */

/* CENTRAL SERVER DECLARES */
DCL VAR(&CSFMT) TYPE(*CHAR) LEN(8) /* Format name */
DCL VAR(&CSFID) TYPE(*CHAR) LEN(4) /* Function identifier */
/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZSCL0100 */
DCL VAR(&CSCNAM) TYPE(*CHAR) LEN(255) /* Unique client name */
DCL VAR(&CSLUSR) TYPE(*CHAR) LEN(8) /* License users handle */
DCL VAR(&CSPID) TYPE(*CHAR) LEN(7) /* Product identification */
DCL VAR(&CSFID) TYPE(*CHAR) LEN(4) /* Feature identification */
DCL VAR(&CSRID) TYPE(*CHAR) LEN(6) /* Release identification */
DCL VAR(&CSTYPE) TYPE(*CHAR) LEN(2) /* Type of information req */
/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZSCS0100 */
DCL VAR(&CSCNAM) TYPE(*CHAR) LEN(255) /* Unique client name */
DCL VAR(&CSCMTY) TYPE(*CHAR) LEN(255) /* Community name */
DCL VAR(&CSNODE) TYPE(*CHAR) LEN(1) /* Node type */
DCL VAR(&CSNNAM) TYPE(*CHAR) LEN(255) /* Node name */
/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZSCN0100 */
DCL VAR(&CSFROM) TYPE(*CHAR) LEN(4) /* From CCSID */
DCL VAR(&CSTO) TYPE(*CHAR) LEN(4) /* To CCSID */
DCL VAR(&CSCTYP) TYPE(*CHAR) LEN(2) /* Type of conversion */
/* DATABASE SERVER DECLARES */
DCL VAR(&DBFMT) TYPE(*CHAR) LEN(8) /* Format name */
DCL VAR(&DBFID) TYPE(*CHAR) LEN(4) /* Function identifier */

/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0100 */
DCL VAR(&DBFILE) TYPE(*CHAR) LEN(128) /* File name */
DCL VAR(&DBLIB) TYPE(*CHAR) LEN(10) /* Library name */
DCL VAR(&DBMBR) TYPE(*CHAR) LEN(10) /* Member name */
DCL VAR(&DBAUT) TYPE(*CHAR) LEN(10) /* Authority to file */
DCL VAR(&DBBFIL) TYPE(*CHAR) LEN(128) /* Based on file name */
DCL VAR(&DBBLIB) TYPE(*CHAR) LEN(10) /* Based on library name */
DCL VAR(&DBOFIL) TYPE(*CHAR) LEN(10) /* Override file name */
DCL VAR(&DBOLIB) TYPE(*CHAR) LEN(10) /* Override libraryname */
DCL VAR(&DBOMBR) TYPE(*CHAR) LEN(10) /* Override membername */

/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0200 */
DCL VAR(&DBNUM) TYPE(*CHAR) LEN(4) /* Number of libraries */
DCL VAR(&DBLIB2) TYPE(*CHAR) LEN(10) /* Library name */

/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAQ0100 */
DCL VAR(&DBSTMT) TYPE(*CHAR) LEN(18) /* Statement name */
DCL VAR(&DBCRRSR) TYPE(*CHAR) LEN(18) /* Cursor name */
DCL VAR(&DBOPT) TYPE(*CHAR) LEN(2) /* Prepare option */
DCL VAR(&DBATTR) TYPE(*CHAR) LEN(2) /* Open attributes */
DCL VAR(&DBPKG) TYPE(*CHAR) LEN(10) /* Package name */
DCL VAR(&DBPLIB) TYPE(*CHAR) LEN(10) /* Package library name */
DCL VAR(&DBDRDA) TYPE(*CHAR) LEN(2) /* DRDA(R) indicator */
DCL VAR(&DBCMT) TYPE(*CHAR) LEN(1) /* Commit control level*/
DCL VAR(&DBTEXT) TYPE(*CHAR) LEN(512) /* First 512 bytes of stmt */

/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAR0100 */
DCL VAR(&DBLIBR) TYPE(*CHAR) LEN(20) /* Library name */
DCL VAR(&DBRDBN) TYPE(*CHAR) LEN(36) /* Relational Database name */
DCL VAR(&DBPKGR) TYPE(*CHAR) LEN(20) /* Package name */
DCL VAR(&DBFLTR) TYPE(*CHAR) LEN(256) /* File name (SQL alias) */
DCL VAR(&DBMBRR) TYPE(*CHAR) LEN(20) /* Member name */
DCL VAR(&DBFFMT) TYPE(*CHAR) LEN(20) /* Format name */

/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAR0200 */
DCL VAR(&DBPLIB) TYPE(*CHAR) LEN(10) /* Primary key table lib */
DCL VAR(&DBPTBL) TYPE(*CHAR) LEN(128) /* Primary key table */
DCL VAR(&DBFLIB) TYPE(*CHAR) LEN(10) /* Foreign key table lib */
DCL VAR(&DBFTBL) TYPE(*CHAR) LEN(128) /* Foreign key table */

/* REMOTE COMMAND SERVER DECLARES */

```



```

CHGVAR VAR(&DQROP) VALUE(%SST(&REQUEST 76 2))
CHGVAR VAR(&WRKLEN) VALUE(%SST(&REQUEST 78 5))
CHGVAR VAR(&DQKLEN) VALUE(&WRKLEN)
CHGVAR VAR(&DQKEY) VALUE(%SST(&REQUEST 83
&DQKLEN))

```

```

/* REMOTE SQL */
CHGVAR VAR(&RSOBJ) VALUE(%SST(&REQUEST 31 10))
CHGVAR VAR(&RSLIB) VALUE(%SST(&REQUEST 41 10))
CHGVAR VAR(&RSCMT) VALUE(%SST(&REQUEST 51 1))
CHGVAR VAR(&RSMODE) VALUE(%SST(&REQUEST 52 1))
CHGVAR VAR(&RSCID) VALUE(%SST(&REQUEST 53 1))
CHGVAR VAR(&RSSSTN) VALUE(%SST(&REQUEST 54 18))
CHGVAR VAR(&RSRSU) VALUE(%SST(&REQUEST 72 4))
CHGVAR VAR(&RSREQ) VALUE(%SST(&REQUEST 76 1925))

```

```

/* NETWORK PRINT SERVER */
CHGVAR VAR(&NPFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&NPFID) VALUE(%SST(&REQUEST 29 4))

```

```

/* IF FORMAT IS SPLF0100 */
IF COND(&NPFMT *EQ 'SPLF0100') THEN(DO)
CHGVAR VAR(&NPJOB#) VALUE(%SST(&REQUEST 33 10))
CHGVAR VAR(&NPUSRN) VALUE(%SST(&REQUEST 43 10))
CHGVAR VAR(&NPJOB#) VALUE(%SST(&REQUEST 53 6))
CHGVAR VAR(&NPFILE) VALUE(%SST(&REQUEST 59 10))
CHGVAR VAR(&NPFIL#) VALUE(%SST(&REQUEST 69 4))
CHGVAR VAR(&NPLEN) VALUE(%SST(&REQUEST 73 4))
CHGVAR VAR(&DECLEN) VALUE(%BINARY(&NPLEN 1 4))
CHGVAR VAR(&NPDATA) VALUE(%SST(&REQUEST 77
&DECLEN))
ENDDO

```

```

/* DATA QUEUE SERVER */
CHGVAR VAR(&DQFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&DQFID) VALUE(%SST(&REQUEST 29 4))
CHGVAR VAR(&DQOOBJ) VALUE(%SST(&REQUEST 33 10))
CHGVAR VAR(&DQOLIB) VALUE(%SST(&REQUEST 43 10))
CHGVAR VAR(&DQOROP) VALUE(%SST(&REQUEST 53 2))
CHGVAR VAR(&DQOLEN) VALUE(%SST(&REQUEST 55 4))
CHGVAR VAR(&DQOKEY) VALUE(%SST(&REQUEST 59 256))

```

```

/* CENTRAL SERVER */
CHGVAR VAR(&CSFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&CSFID) VALUE(%SST(&REQUEST 29 4))

```

```

/* IF FORMAT IS ZSCL0100 */
IF COND(&CSFMT *EQ 'ZSCL0100') THEN(DO)
CHGVAR VAR(&CSCNAM) VALUE(%SST(&REQUEST 33 255))
CHGVAR VAR(&CSLUSR) VALUE(%SST(&REQUEST 288 8))
CHGVAR VAR(&CSPID) VALUE(%SST(&REQUEST 296 7))
CHGVAR VAR(&CSFID) VALUE(%SST(&REQUEST 303 4))
CHGVAR VAR(&CSRID) VALUE(%SST(&REQUEST 307 6))
CHGVAR VAR(&CSTYPE) VALUE(%SST(&REQUEST 313 2))
ENDDO

```

```

/* IF FORMAT IS ZSCS0100 */
IF COND(&CSFMT *EQ 'ZSCS0100') THEN(DO)
CHGVAR VAR(&CSCNAM) VALUE(%SST(&REQUEST 33 255))
CHGVAR VAR(&CSCMTY) VALUE(%SST(&REQUEST 288 255))
CHGVAR VAR(&CSNODE) VALUE(%SST(&REQUEST 543 1))
CHGVAR VAR(&CSNNAM) VALUE(%SST(&REQUEST 544 255))
ENDDO

```

```

/* IF FORMAT IS ZSCN0100 */
IF COND(&CSFMT *EQ 'ZSCN0100') THEN(DO)
CHGVAR VAR(&CSFROM) VALUE(%SST(&REQUEST 33 4))
CHGVAR VAR(&CSTO) VALUE(%SST(&REQUEST 37 4))
CHGVAR VAR(&CSCTYP) VALUE(%SST(&REQUEST 41 2))

```

ENDDO

```
/* DATABASE SERVER */
CHGVAR VAR(&DBFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&DBFID) VALUE(%SST(&REQUEST 29 4))
/* IF FORMAT IS ZDAD0100 */
IF COND(&CSFMT *EQ 'ZDAD0100') THEN(DO)
CHGVAR VAR(&DBFILE) VALUE(%SST(&REQUEST 33 128))
CHGVAR VAR(&DBLIB) VALUE(%SST(&REQUEST 161 10))
CHGVAR VAR(&DBMBR) VALUE(%SST(&REQUEST 171 10))
CHGVAR VAR(&DBAUT) VALUE(%SST(&REQUEST 181 10))
CHGVAR VAR(&DBBFIL) VALUE(%SST(&REQUEST 191 128))
CHGVAR VAR(&DBBLIB) VALUE(%SST(&REQUEST 319 10))
CHGVAR VAR(&DBOFIL) VALUE(%SST(&REQUEST 329 10))
CHGVAR VAR(&DBOLIB) VALUE(%SST(&REQUEST 339 10))
CHGVAR VAR(&DBOMBR) VALUE(%SST(&REQUEST 349 10))
ENDDO
```

```
/* IF FORMAT IS ZDAD0200 */
IF COND(&CSFMT *EQ 'ZDAD0200') THEN(DO)
CHGVAR VAR(&DBNUM) VALUE(%SST(&REQUEST 33 4))
CHGVAR VAR(&DBLIB2) VALUE(%SST(&REQUEST 37 10))
ENDDO
```

```
/* IF FORMAT IS ZDAQ0100 */
IF COND(&CSFMT *EQ 'ZDAQ0100') THEN DO
CHGVAR VAR(&DBSTMT) VALUE(%SST(&REQUEST 33 18))
CHGVAR VAR(&DBCRRS) VALUE(%SST(&REQUEST 51 18))
CHGVAR VAR(&DBSOPT) VALUE(%SST(&REQUEST 69 2))
CHGVAR VAR(&DBATTR) VALUE(%SST(&REQUEST 71 2))
CHGVAR VAR(&DBPKG) VALUE(%SST(&REQUEST 73 10))
CHGVAR VAR(&DBPLIB) VALUE(%SST(&REQUEST 83 10))
CHGVAR VAR(&DBDRDA) VALUE(%SST(&REQUEST 93 2))
CHGVAR VAR(&DBCMT) VALUE(%SST(&REQUEST 95 1))
CHGVAR VAR(&DBTEXT) VALUE(%SST(&REQUEST 96 512))
ENDDO
```

```
/* IF FORMAT IS ZDAR0100 */
IF COND(&CSFMT *EQ 'ZDAR0100') THEN DO
CHGVAR VAR(&DBLIBR) VALUE(%SST(&REQUEST 33 20))
CHGVAR VAR(&DBRDBN) VALUE(%SST(&REQUEST 53 36))
CHGVAR VAR(&DBPKGR) VALUE(%SST(&REQUEST 69 20))
CHGVAR VAR(&DBATTR) VALUE(%SST(&REQUEST 89 20))
CHGVAR VAR(&DBFULR) VALUE(%SST(&REQUEST 109 256))
CHGVAR VAR(&DBMBRR) VALUE(%SST(&REQUEST 365 20))
CHGVAR VAR(&DBFFMT) VALUE(%SST(&REQUEST 385 20))
ENDDO
```

```
/* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAR0200 */
/* IF FORMAT IS ZDAR0200 */
IF COND(&CSFMT *EQ 'ZDAR0200') THEN DO
CHGVAR VAR(&DBPLIB) VALUE(%SST(&REQUEST 33 10))
CHGVAR VAR(&DBPTBL) VALUE(%SST(&REQUEST 43 128))
CHGVAR VAR(&DBFLIB) VALUE(%SST(&REQUEST 171 10))
CHGVAR VAR(&DBFTBL) VALUE(%SST(&REQUEST 181 128))
ENDDO
```

```
/* REMOTE COMMAND SERVER */
CHGVAR VAR(&RCFMT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&RCFID) VALUE(%SST(&REQUEST 29 4))
CHGVAR VAR(&RCPGM) VALUE(%SST(&REQUEST 33 10))
CHGVAR VAR(&RCLIB) VALUE(%SST(&REQUEST 43 10))
CHGVAR VAR(&RCNUM) VALUE(%SST(&REQUEST 53 4))
CHGVAR VAR(&RCDATA) VALUE(%SST(&REQUEST 57 6000))
```

```
/* SIGNON SERVER DECLARES */
CHGVAR VAR(&SOFNT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&SOFID) VALUE(%SST(&REQUEST 29 4))
```



```

/* DATABASE INIT */
SQLINIT:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT

/* NATIVE DATABASE */
NDB:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT
/* DATABASE SQL */
SQLSRV:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT

/* RETRIEVE OBJECT INFORMATION */
RTVOBJ:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT

/* DATA QUEUE SERVER */
ODATAQ:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT
/* NETWORK PRINT SERVER */
NETPRT:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT
/* CENTRAL SERVER */
CENTRAL:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT
/* REMOTE COMMAND AND DISTRIBUTED PROGRAM CALL */
RMTCMD:

/* IN THIS CASE IF A USER ATTEMPTS TO DO A REMOTE COMMAND AND DISTRIBUTED */
/* PROGRAM CALL AND HAS A USERID OF userid THEY WILL NOT BE ALLOWED TO */
/* CONTINUE. */
IF COND(&USER *EQ 'userid') THEN(CHGVAR VAR(&STATUS) VALUE('0'))

GOTO EXIT
/* SIGNON SERVER */
SIGNON:

/* SPECIFIC LOGIC GOES HERE */

GOTO EXIT

EXIT:
ENDPGM

```

Examples: Create QIBM_QZDA_INIT exit programs with CL commands

You can create IBM i QIBM_QZDA_INIT exit programs using CL commands.

The following example illustrates how to set up a QIBM_QZDA_INIT user exit program with control language (CL) commands.

Note: Read the Code example disclaimer for important legal information.

```

/*****
/* IBM i - Sample User Exit Program */
/* */
/* Exit Point Name : QIBM_QZDA_INIT */

```

```

/*
/* Description      : The following Control Language program
/*                   handles ODBC security by rejecting requests
/*                   from users who use ODBC and signon using a
/*                   user profile of 'GUEST'. It is a shell
/*                   program for developing exit programs
/*                   tailored for your environment.
/*
/*****
PGM PARM(&FLAG &REQUEST)

```

```

/*****
/* Program call parameter declarations
/*
/*****
DCL VAR(&FLAG) TYPE(*CHAR) LEN(1)
DCL VAR(&REQUEST) TYPE(*CHAR) LEN(285)
/*****
/* Parameter declares for Request Format
/*
/*****
DCL VAR(&USER) TYPE(*CHAR) LEN(10) /* User profile
DCL VAR(&SRVID) TYPE(*CHAR) LEN(10) /* Server Id (*SQL)
DCL VAR(&FORMAT) TYPE(*CHAR) LEN(10) /* Format ZDAI0100
DCL VAR(&FUNC) TYPE(*CHAR) LEN(4) /* Function Id 0
DCL VAR(&INTTYP) TYPE(*CHAR) LEN(63) /* Interface type
DCL VAR(&INTNAM) TYPE(*CHAR) LEN(127) /* Interface name
DCL VAR(&INTLVL) TYPE(*CHAR) LEN(63) /* Interface level

```

```

/*****
/* Extract the various parameters from the structure.
/*
/*****
CHGVAR VAR(&USER) VALUE(%SST(&REQUEST 1 10))
CHGVAR VAR(&SRVID) VALUE(%SST(&REQUEST 11 10))
CHGVAR VAR(&FORMAT) VALUE(%SST(&REQUEST 21 8))
CHGVAR VAR(&FUNC) VALUE(%SST(&REQUEST 29 4))
CHGVAR VAR(&INTTYP) VALUE(%SST(&REQUEST 33 63))
CHGVAR VAR(&INTNAM) VALUE(%SST(&REQUEST 96 127))
CHGVAR VAR(&INTLVL) VALUE(%SST(&REQUEST 223 63))

```

```

/*****
/* Set return code to allow the request.
/*
/*****
CHGVAR VAR(&FLAG) VALUE('1')
/*****
/* If interface type is ODBC and User is 'GUEST' reject the
/* signon attempt.
/*
/*****
IF ((%SST(&INTTYP 1 4) *EQ 'ODBC') *AND +
    (&USER = 'GUEST ')) THEN(DO)

    /*****
    /* Set return code to NOT allow the request.
    /*
    /*****
    CHGVAR VAR(&FLAG) VALUE('0')
    ENDDO
ENDPGM

```

Examples: Create QIBM_QZDA_INIT exit programs with ILE C

You can create IBM i QIBM_QZDA_INIT exit programs using ILE C.

The following example illustrates how to set up a QIBM_QZDA_INIT user exit program with ILE C..

Note: Read the Code example disclaimer for important legal information.

```

/*****
/* IBM i - Sample User Exit Program
/*
/*
/* Exit Point Name : QIBM_QZDA_INIT
/*
/* Description      : The following ILE C language program
/*                   handles ODBC security by rejecting requests
/*                   from users who use ODBC and signon using a
/*                   user profile of 'GUEST'. It is a shell
/*                   program for developing exit programs
/*                   tailored for your environment.
/*
/*****
#include <stdio.h>

```

```

#include <string.h>
#include <ezdaep.h>
main(int argc, char *argv[])

{
Qzda_Init_Format_t input;          /* input format */

/*****
/* Copy format parameter to local storage. */
/*****
memcpy(&input,(Qzda_Init_Format_t *) argv[2],
sizeof(Qzda_Init_Format_t));
/*****
/* If user profile is 'GUEST' and interface type is 'ODBC'
/* reject the connection. */
/*****
if (memcmp(input.User_Profile,"GUEST",10)==0 &&
memcmp(input.Interface_Type,"ODBC",4) == 0)

/*****
/* Reject the connection. */
/*****
strcpy(argv[1],"0");
else
/*****
/* Allow the connection. */
/*****
strcpy(argv[1],"1");
return;
}

```

Console Advanced Topics

This section includes advanced topics which may help administrators when setting up and managing the system console.

Console planning considerations

Take into consideration the following points when performing Console-related activities.

The following information applies to all systems:

- If no console type is specified, such as when a new logical partition is being created, the supporting hardware IOP specified during the creation process takes precedence. If the available hardware contains more than one adapter for a selected console type, then the first console workstation to connect will be configured to become the console.
- There is also a special set of console options called **Allow console recovery** and **console can be taken over by another console**. This set of functions allows 5250 Console to take control from another console device. The default value of this console option is disabled.
 - When the option is enabled:
 - The first 5250 Console device connected becomes the console. Additional LAN-connected 5250 Console devices will have a special DST sign-on.
 - All other 5250-based connections will have the new Console Information Status window.
 - Console recovery without the loss of the job is available.
 - When the option is disabled:
 - All 5250-based connections will be presented the new Console Information Status window.
 - Console recovery without the loss of the job is not available.
- 5250 Console allows multiple LAN connections to a single system or logical partition, but only one 5250 session can have control of a system at a time. An active console is the command interface to a system through 5250 emulation that is currently interacting with the system. More than one 5250 Console device might have data on the screen but only one is actually active.

- The client PC also allows multiple local console connections to the system configuration for a single PC.
- Try to avoid putting your 5250 Console on the same IOP as the storage devices.
 - There might be configurations when this cannot be avoided.
 - During very heavy usage of storage devices, the console might appear to stop working temporarily, but it should resume operation shortly. If the console is placed on the same IOP as the storage devices, enable the console options **Allow console recovery** and **console can be taken over by another console**.

The following information pertains to console takeover and recovery:

For IBM i there is a special set of functions known as console takeover and recovery, which allows a LAN-connected 5250 Console to take control from another console device. Use the following information to help determine what console devices are best for your work environment and how to deploy these devices to take advantage of these functions.

- **Takeover** is the process used for a LAN-connected console device to take control from the current console device. The user signed on to the PC that wishes to take control requires special authority and is initiating the takeover from a new menu.
- **Recovery** is the process of regaining control of the job running on the console after a problem with the console was encountered. The recovery process may be to the same console device or a different console device and may be facilitated by additional work to enable a device using a different connectivity.

When the takeover option is enabled and each console capable device running 5250 emulation is properly connected, a screen of data is presented regardless of whether it is the active console. In V5R4 and later releases, more than one device will have data on the screen after the console has been established. This results in no blank console screens showing *Disconnected* when initially connecting a device. The new function now allows the job at the console to be transferred to another device without causing loss of data.

This function is accomplished by suspending the data stream to a console that loses a connection or is being taken over, save further data and then send that data to the next device to become the console, even if the device is the same former console. Recoverability is essentially taking over the console from the same or different qualified device regardless of what the former console was doing.

The following information pertains to nonpartitioned partitions:

- Console supporting hardware may be required to be located in specific slot locations, based on model.

The following information pertains to a multi-partitioned environment:

- If you plan to use 5250 Console as your primary console or as a backup console, you must tag the IOA to support the primary console.
- When more than one console adapter is available for a single IOP, the adapter with the lowest bus address is chosen for use by 5250 Console. For example, you tag an IOP that has two LAN adapters installed. The system uses the first adapter found on the bus. However, during an IPL the first adapter might not be ready in time and the system can select the second adapter. This process can prevent the console from starting immediately or you might not be able to use that resource for your intended purposes. It is recommended that you install only one console-capable adapter that matches your configurations for a single IOP or tag the IOA instead.

Planning considerations for your backup console

Most system plans include a level of redundancy to allow for the event of hardware failures; however, many people do not consider the console in their plans. To recover quickly from an unexpected loss of your console, you need to plan for a backup console.

Considerations for a backup console

- The adapter location is fixed, or at least limited, for independent systems. Based on your system hardware requirements, you might have limited choices of console types. Try to accommodate at least one additional console type, if possible.
- Consider using the take over and recovery option as part of your backup console strategy. However, the hardware used for the new console type must exist and be available at the time of recovery.
- If you are working in a partitioned environment, consider:
 - Tagging an IOP that has two similar console adapters reporting to it. For example, two 2849 adapters for the same IOP can sometimes make it difficult to determine, in advance, which adapter is for the console. When tagging the IOP, make sure that it only has one console-capable adapter per connectivity (for example, only one 2849). Each adapter can support a different console type but only one adapter type is present. The lowest addressed adapter on the bus is attempted first. But if that adapter is slow in reporting to the system, another adapter might get selected instead, when two adapters of the same connectivity are present. Another example might be that the IOP has both a 2838 and a 2849 Ethernet adapter reporting to it. They are different adapters but have the same connectivity for the console.
 - Consider a shared resource environment in which you can allocate and deallocate a console supporting IOP to a partition on a part-time basis. Many work environments rarely need a console device on a full-time basis and you can reduce your initial cost of dedicated hardware by implementing this concept.
 - If the load source storage device fails and the system recovery includes the use of the IBM distribution Licensed Internal Code media instead of a customer backup, and the system is using Operations Console (LAN), you might have to use another console type for the initial portion of the system recovery.

Configuration types for backup consoles

When planning the configuration of the backup console or consoles remember that recovering from the loss of the console depends on many factors. Some of these factors include, the model and series, the hardware resources available, the previous console type, and the intended console type. Recovery might consist of repairing the currently failed console or temporarily replacing it with another console type. Most changes of a console type can be performed without the need for an IPL but there may be circumstances in which an IPL will still be necessary. When using the console service functions, console-supporting hardware must be installed and available prior to performing the function. Any partition tagging of resources must also have been done already.

Important: If you plan to use 5250 Console on a LAN as a backup to another console type, the network adapter must be located in a console designated slot or in a properly tagged IOP or IOA.

Backup console configuration considerations

- If you access your system remotely, consider off-site console capability or another type of connectivity for the console. A local console on a network can be backed up with an additional local console on a network PC.
- In a logical partition or multiple-system environment, you will most likely be using multiple local consoles on a network (LAN) configuration on a single PC as your primary consoles. Consider additional PCs using this same type configuration. Avoid supporting too many consoles on the same PC if possible. The PC resources can be easily overwhelmed when supporting multiple consoles and remote control panels.

- Consider multiple local console on a network configurations in large environments so that each PC has a core set of console responsibilities and the overlap coverage of backup configurations with each other. For example, if you have a PC that supports 10 local consoles on a network configuration and another PC with the same number of primary consoles for another 10 partitions, instead of backing up each PC with the other's configuration, you add a third PC and spread the 20 consoles out so that two PCs back up a portion of each PC's primary console configurations. Another consideration is a dedicated PC to be the backup of a certain number of consoles, but not connected until necessary.

In summary, consider incorporating as much redundancy as possible into your console configuration. You can reduce your exposure to a catastrophic console failure by using another method to provide a console in place, or by making compromises and adjustments for the various hardware requirements necessary to overcome the various levels of failures.

Console hardware requirements

You need to understand the hardware requirements to configure a Console.

If you want to use an 5250 Console local console on a network (LAN), you need to install the LAN adapter for Operations Console according to your model. To find out the specifics of your system, you can use the Display Hardware Resources (DSPHDWRSC) command. Table 1 shows the supported cards for LAN connectivity.

Card name or number	Description
2744	PCI 100 Mbps Token ring Adapter
2838	PCI 10/100 Mbps Ethernet IOA
2849	PCI 10/100 Mbps Ethernet IOA
Host Ethernet Adaptor (HEA) ³	PCI 10/100/1000 Mbps or 10 Gbps Ethernet
57xx ³	PCI 10/100/1000 Mbps or 10 Gbps Ethernet
268C	Virtual Ethernet adapter (VLAN)

Notes:

1. LAN console is supported on Virtual Ethernet Adapter (VLAN). A virtual Ethernet adapter is defined on Host Ethernet Adaptor (HEA) or using VIOS.
2. LAN console is not supported with line descriptions which are configured for Ethernet Link Aggregation.
3. See [Hardware requirements for Ethernet](#) for supported adapters and speeds.

Related information

[IBM Systems Support Web site](#)

[Display Hardware Resources \(DSPHDWRSC\) command](#)

[TCP/IP techniques to connect virtual Ethernet to external LANs](#)

[Hardware requirements for Ethernet](#)

Planning considerations for your Operations Console installation or upgrade

To plan for your Operations Console installation or upgrade, you need to know this information.

Prerequisite information for Operations Console users upgrading to or installing i 7.2

When you receive the IBM i release upgrade, all of the user IDs included with the operating system are expired, except for 11111111 (eight 1's). For all upgrades and installations, you need to establish

a connection between the IBM i platform and Operations Console using the service tools user ID of 11111111. This prevents any failed re-authentication of the client connection to the system caused by expired user IDs. This is especially important for automated installations and upgrades.

Note: Failure to comply with the above actions may prevent the console from working correctly during the upgrade or installation.

Important: During a manual IPL of the system and if no console has been specified before, you receive two additional displays to confirm the setting of the console type. The first requires that you accept your current console type, and the second shows that a value did not previously exist (a zero is present for the old value) and the new value is shown. Press Enter to exit and set the console type automatically. The IPL then continues to the IPL or Install the System display. This condition is most likely to occur during the installation of a new partition, but it can happen on your first manual IPL, for example, during the A-mode IPL following the restore of Licensed Internal Code, or during the upgrade or installation when a console value of zero is found.

Migration to Operations Console before your system upgrade

When migrating a former load source device that was used with Operations Console local console on a network configuration, and this load source device is to become a load source device in a new partition, the LAN adapter must be deallocated before removing the device from the old partition and installing it in the new partition.

If the result of your system upgrade includes a Hardware Management Console (HMC), make sure to use this as the system console until your upgrade is complete. Because some upgrades are staged, your intended console might lack the necessary resources for a period of time. Because the HMC uses virtual resources, you will not have problems with a lack of resources for another console type.

Related information

[Installing, upgrading, or deleting IBM i and related software](#)

Preparation for your network environment

To prepare for your network environment, you need to identify and comply with the minimum network configuration required to set up an 5250 Console local console on a network (LAN) configuration.

If your system is new and you chose a local console on a network configuration, the adapter is already allocated for use by the system. The LAN adapter is dedicated for service tools. It is suggested that you restrict LAN topologies for LAN-attached local consoles to a single, physical ring, hub, switch, or router environment. If the local console on a network is used in a larger network topology, it is suggested that you use broadcast (DHCP) packet filtering. This might be as simple as connecting the PC and system using an inexpensive hub or router. Temporarily, you could also use an Ethernet cable (when using a 2849 adapter). When you have only a single PC or few devices connected to the system using a hub, and these devices do not connect to another network or the Internet, you can then use any numeric numbers for addresses, for example, *1.1.1.x* or *10.220.215.x* (where *x* can be 2 through 254, but avoid *x.x.x.1*, which might cause problems in some hubs). However, if you have a network that many users share, or in which the devices are connected to the Internet, consult a network administrator for addresses.

Network security

Implementing some sort of physical security of the console over a LAN connection is critical. For instance, consider configuring a local console on a network separate from the main network (or the company intranet) and strictly controlling access to the PC that acts as the console.

Security of your Operations Console configuration

Operations Console security consists of user authentication, data privacy, and data integrity.

User authentication security is required to sign on to the console display.

The IBM i console security consists of:

User authentication

This security provides assurance as to who is using the service device. All problems related to user authentication are the same regardless of console type. For more information, see the Service tools topic.

Data privacy

This security provides confidence that the console data can only be read by the intended recipient. If the physical connection is secure as discussed under service device authentication, the console data remains protected. To protect the data, ensure that only authorized people enter the computer room.

Operations Console local console on a network uses a secure network connection.

Data integrity

This security provides confidence that the console data has not changed en route to the recipient. If the physical connection is secure, the console data remains protected. An Operations Console local console on a network uses a secure network connection.

Data encryption

Enhanced authentication and data encryption provide network security for console procedures. 5250 Console on a network uses TLS.

Administration

Operations Console administration allows system administrators to control access to console functions, including the remote control panel.

Important: Consider the following situations when administering Operations Console local console over a network:

- For the remote control panel, mode selections require security authorization for the user that authenticates the connection, such as that provided by QSECOFR. Mode selections include Manual and Normal. Also, when connecting the remote control panel using a network, the service tools user ID must have authority to the control panel data on the system or on the partition that the remote control panel connects to.
- If you implement a network security tool that probes ports for intrusion protection, be aware that Operations Console uses ports 449, 2300, 2323, 3001, and 3002 for normal 5250. If your tool probes any of these ports, it might cause loss of the console, which might result in an IPL to recover. Exclude these ports from intrusion protection tests.

Console control features

If IBM i Operations Console detects that the same user (which means the same PC IP address and the same User ID) and with the Skipped setting enabled, the normal Takeover Sign on screen will also be skipped. If this feature is not desired, it can be disabled by setting the option to Show.

Protection tips

When using an Operations Console local console on a network, change your password for the following DST user IDs: QSECOFR, 11111111, 22222222, and QSRV.

Related information

[Service tools user IDs and passwords](#)

Scenario: Understanding your configuration

This scenario helps you understand the configuration for your environment.

Use the following scenario to help assist you in understanding your Operations Console configuration. Keep in mind that this scenario applies only to nonpartitioned systems.

Scenario: Consoles for multiple systems or partitions

This scenario discusses a situation in which you want to manage multiple systems or partitions.

Your company owns an IBM i product, and you want to use the PC to manage your system. You need to manage multiple systems or partitions from one console. You have a secured network that you can configure your console on.

For this scenario, configure a local console on a network.

Advantages

- You can configure a single PC to be the console for several different systems or partitions as long as they are connected to the service connection network.
- The administrator does not need to be physically near the system to manage the console.
- Security features are available to protect your console connections.
- A local console on a network is the connectivity of choice for partitions in an LPAR environment.
- Multiple PCs can be configured as a console to a system or partition, but only one PC can act as an active console at a time.

Disadvantages

- No console is available when a network failure takes place unless a backup console is available.
- Your system requires a separate LAN card to be used by the console or other service tools.

Related information

[Logical partitions](#)

Managing Operations Console

You can manage your Operation Console by performing tasks such as changing the console configuration, switching between different console types, and changing passwords.

Takeover or recovery of an Operations Console connection

You can use these functions to take control of another console device.

A special set of functions included in IBM i allow an Operations Console to take control from another console device. The two main actions that you can do are:

- **Takeover** is the process used for a LAN-connected console-capable device to take control from the current LAN-connected console device.
- **Recovery** is the process of regaining control of the job running on the console after a problem with the console was encountered. The recovery process may be to the same console device or a different console device and may be facilitated by additional work to enable a device using a different connectivity.

Every console-capable device running 5250 emulation, regardless of connectivity, will be presented a window of data regardless of whether it is the console when it successfully connects. This means that more than one device will have data on the window after the console has been established. A console device will not have a blank window showing `Disconnected`. This action allows the job at the console to be "transferred" to another device without causing loss of data. When the console option for takeover is enabled, the system also has enhanced recoverability from loss of the console.

The recovery action is accomplished by suspending the data stream to a console that loses a connection, or is being taken over, saving further data and then sending that data to the next device to become the console, even if the device is the same former console. Recoverability is essentially just taking over the console from the same or different qualified device regardless of what the former console was doing.

The default setting for the console take over and recovery function is **disabled**. If this function is disabled, all console-capable devices will open the Console Information Status window whenever they are not the active console.

Benefits of these functions include convenience and redundancy. Console-capable devices can be placed around a site, or multiple sites, allowing users to move around and gain control of the system from any of these devices. Whatever the previous console's activity was, the new console is at the exact same place, even during the process of restarting the system or installing the IBM i operating system. When the console option for takeover is enabled, the system also has enhanced recoverability from the loss of the console.

Takeover details

Here is the additional information about the console takeover function.

- This function must be enabled if you want to take over the console or be protected from the loss of the console using the recovery function.
- The option **Allow console recovery and console can be taken over by another console** is enabled on the Select Console window in DST or SST.
- Any 5250 emulation-based console device can be used to recover a loss of the console by changing the console type. This may require a reallocation of hardware to support the new console type.
- The DST user ID used to sign on at an eligible device must also have the user privilege of the takeover console.
- Only devices with the same attributes (for example: 24x80 or 27x132) can perform a takeover. For example, if device LAN1 is running in 24x80 mode and LAN2 is running in 27x132 mode, and LAN1 is the console, LAN2 will see **NO** in the **Take over the console** field.
- Data on the Console Information Status screen does not change. Currently, there is no method to automatically refresh the data. You can manually cause a refresh of all fields except the Take over the console field by pressing Enter. The user would have to exit this screen and sign on again to see a change to that field.
- Takeover is supported in a D-mode IPL. Two devices can be connected, with data, at the same time during a D-mode IPL.

Related reference

Recovery details

Here is the additional information about the console recovery function.

Recovery details

Here is the additional information about the console recovery function.

- The recovery of the console using a device with the same console connectivity is directly tied to the takeover option. If you do not want takeover capability but do want recovery from loss of the console, you must still enable the takeover option.
- Console recovery utilizes the takeover function. Recovery can be from the same device or another device of the same connectivity. For example, if you are using 5250 Console and have multiple PCs set up to be the console and the existing console fails you can use the takeover function from the same PC, after correcting the reason for the failure, or from another PC. Regardless of what the former console was doing, the new console will be in the same job, at the same step, as the original console would have been. The job continues even though the console was not operational.
- The recoverability of the console using a different console connectivity provides additional options for the user. If you have a backup console plan that involves the need for a change in the console type, consider the following:
 - To reduce the number of steps necessary to accomplish a recovery, place all console supporting adapters for use by the same IOP.
 - The console type change can be immediate, if the system is not partitioned and depending on the method used to change the console. Examples would be:

- Using DST or SST to change the console type would allow the console to change only at the next IPL. Following this action with a force DST (function 21) using the control panel might not always work either.
- Changing tags in the partitioned environment requires an IPL to cause a change of the console.
- However, using the console service functions on a nonpartitioned system forces the system to do this search and activates the appropriate hardware and tasks immediately.
- Supporting hardware for each console type you want to use for recovery must be available at the time of recovery. You also need to change the console type, either using an available menu or using the console service functions.

To accomplish a recovery using a different console type, you must set the new console type **before** attempting the takeover. This means that either the supporting hardware must already be available, including any logical partition tagging, or you must move supporting hardware, physically or logically, before attempting the recovery. Then, you will have to use one of the methods to change the console type to the desired setting. You can use an existing workstation and SST, if available, the native macro, or the console service functions.

- If you change the console type value during a D-mode IPL, you should be able to connect another device as long as the new console type has supporting hardware and a device.

Related reference

Takeover details

Here is the additional information about the console takeover function.

Using the operations console service functions

Operations console service functions are the emergency operations console recovery functions. This section gives an overview of the functions and details on how the operations console service function works.

Enabling console takeover

Before you can enable console takeover, you must have the takeover console privilege.

Use the procedure following to grant takeover console privilege and then continue with the next procedure to enable console takeover.

To add the takeover console privilege to a user, perform the following steps:

Note: To perform any of the following procedures using SST, select the option Work with service tools user IDs and Devices everywhere it says select Work with DST environment and skip the step Select **System Devices**.

1. Access Dedicated Service Tools (DST) or System Service Tools (SST).
2. Select **Work with DST environment**.
3. Select **Service tools user IDs**.
4. Type a 7 on the line in front of the desired user ID and press Enter.
5. Scroll down until you find the option **Take over console** and place a 2 on that line to grant the user this privilege and press Enter.

To repeat this procedure for additional user IDs, repeat steps 4 and 5.

This privilege will be used the next time the user ID signs on.

Note: When a user signs on at a device capable of taking over the console the status of the takeover the console field is updated. In order to reflect a change, such as a user being granted the takeover console privilege, the user would have to exit the Console Information Status window, using F3 or F12 and sign on again.

To enable the console takeover and recovery, perform the following:

- a) Access Dedicated Service Tools (DST) or System Service Tools (SST).
- b) Select **Work with DST environment**.

- c) Select **System devices** (skip this step if using SST).
- d) Select **Select Console**.
- e) Type a 1 in the option column for **Allow console recovery and console can be taken over by another console** and press Enter.

The console takeover option will take effect immediately.

Scenario: Takeover and recovery

This scenario can help you understand the takeover and recovery options.

Scenario: LAN connected devices only with takeover enabled

This scenario describes what happens during an IPL when console takeover is enabled and more than one LAN-connected device is available.

The LAN devices are called LAN1, LAN2, and LAN3. The IPL is being performed in unattended (Normal) mode.

At the point in an IPL when the console device is being determined, it is more or less a race condition if more than one device is connecting at a time. The first device to connect, of the type specified by the console type setting (LAN in our example), becomes the console and will be presented with the usual console screens.

For our example let's say LAN1 is the first device connected. During the IPL this device will show the IPL status changes just like any other console and eventually the IBM i sign on window. LAN2 and LAN3 will show a special DST signon screen with a new line of data stating ATTENTION: This device can become the console. The rest of the window will be the same as any other DST signon window. At LAN2 a user with the user privilege of take over console signs on. This user will now be presented the same Console Information Status screen and the take over the console field will show a YES indicating that takeover is possible. At LAN3 a user without the take over console privilege signs on. The take over the console field will show as NO since the user does not have the correct authority for takeover.

At this point, only one device has met all the conditions for a console takeover. At the bottom of the window is F10 (Take over console connection). Pressing F10 presents the user with the Take over Console Connection From Another User window. This is a confirmation window that gives the user a last chance to cancel the takeover. Selecting 1 and then pressing Enter at this point causes the takeover to occur. Almost immediately, LAN1 gets the special DST sign-on window and LAN2, the device that initiated the takeover, has the exact same window that LAN1 had when the transfer took place. The job, if something was running, does not know that this action took place. In fact, the original console could have been installing Licensed Internal Code or the IBM i operating system, or it could have been running a complete system save in restricted state and the system does not know it. You can even disconnect the console connection and come back later, reconnect, and you can still get the current job's window data. If a large amount of window data was sent by the job and could not be delivered, the data will be stored until later. When a console is reconnected by an authorized user (has the takeover console privilege) from an eligible device, the user might see fast window refreshes until all the stored data has been delivered. Actually, doing a disconnection and a reconnection is considered a recovery (not a takeover).

The data present at LAN3 will not change after the takeover. Currently, there is no method to automatically refresh the data. However, if the user at LAN3 pressed Enter, a manual refresh of all fields except the Take over the console field would occur. The user would have to exit this screen and sign on again to see the change to that field.

Management of multiple consoles

If you have more than one workstation that is capable of being the console to the same system or partition, there might be more than one way of using those devices as the console, depending on your configuration and circumstances.

Multiple PC consoles on a network

When a workstation is already a console and another local console on a network attempts to become the console, the connection is successful and the console attempting to connect is presented with a sign-on window or status window.

Which window you see depends on whether the option **Allow console recovery and console can be taken over by another console** is enabled or not. If the option is enabled, then you are presented with a sign-on window to verify whether you have permission to perform a takeover operation. If the option is not enabled, then you are presented with the Console Information Status window. The reason will be displayed at the bottom of the window. In this case, takeover is not enabled.

The Console Information Status window shows which device is currently the console based on the last successful connection. Many connections of this type can be connected but only one can be the active console. Also, leaving the newly connected PC in this state will not allow the console activities to be automatically transferred to this PC. In this case you have two choices.

- Disconnect the connection using the Operations Console window. To disconnect the connection, follow these steps:
 1. Select the connection name you want to disconnect.
 2. Click **Connection > Disconnect**.
- Disconnect the emulator session. To disconnect the emulator, follow these steps:
 1. In the emulator window, click **Communication**.
 2. Select **Disconnect**.

If no device is acting as the console the next time a connection is made, either through Operations Console or the emulator, this PC will become the console. It is up to you to determine whether or not this method is the correct way to manage console activities.

Changing from one console type to another

Depending on how your console connection is set up, you can change to a different console type.

As part of your migration, you might need to deallocate the LAN adapter from use by the Operations Console. For instructions on deallocating or moving the LAN adapter, see the Deallocating or moving the LAN adapter card topic.

If you are using an IBM i model that uses a Hardware Management Console (HMC), you can change from HMC to LAN Console or from LAN Console to HMC. For further details, see the Managing consoles, interfaces, and terminals > Changing consoles, interfaces, and terminals topic in the IBM Systems Hardware Information Center.

For information regarding the Hardware Management Console (HMC), see the IBM Systems Hardware Information Center.

Related tasks

[Deallocating or moving the LAN adapter card from use by Operations Console](#)

During a migration, you might need to deallocate the LAN card from use by Operations Console. You need to deallocate the LAN card if you are not planning on using an Operations Console local console on a network configuration or the service tools server.

Related information

[Electronic customer support](#)

[IBM Systems Hardware Information Center](#)

Switching from one console type to another when the current console is not operational

If you develop a problem with the current console, you can use one of these methods to change the console type.

- If you are using a LAN-connected PC for the console and have another PC set up to be the console, you might be able to use the other PC for the console until the problem with the first console is fixed.

- Use SST from another workstation.
- Use the console service functions.
- Use the appropriate native macro from another workstation.

Note: Any hardware allocations or configurations will have to be accomplished before connecting with another connectivity. For example, if you are using a shared IOP in a partitioned environment you can deallocate and allocate the resource from one partition to another, if your hardware supports this method. If you planned for a backup console some or all of this work may already be done. If you do not have a backup console in place, some tasks may require one or more IPLs to get the system into a state where you can use the new console.

Assuming that all hardware and configurations have been done, you would disconnect the existing console (LAN-connected); use the console service functions, native macro, or SST from another workstation to set the console type. Then using the console service functions or native macro OPSCONSOLE RESTART automatically activates the adapter.

Note: There are several console service functions that might be needed to recover or debug an Operations Console problem depending on the problem, the connectivity used for the current console, the target console type, and the current state of the system. If you are unsure of any function or recovery action, contact your service provider for assistance.

Related reference

Using the operations console service functions

Operations console service functions are the emergency operations console recovery functions. This section gives an overview of the functions and details on how the operations console service function works.

Managing your local console on a network

After you configure a local console on a network, you can manage your console.

Starting the system using a manual IPL

You can start your system by performing a manual initial program load (IPL) using the system's physical control panel.

These instructions assume the system is powered off. If the system is powered on, use one of the various methods available to start the manual IPL.

To perform a manual IPL, follow these steps:

1. Look at the Function/Data display on the control panel.
The Function/Data display should show Manual and 01 B.
2. If the system is in Manual mode and will IPL on the B side, then go to step 8. If the system is not in Manual mode or not set to IPL on the B side, then continue with step 3.
3. If the Function/Data display is lit, then, continue with step 4.
If the Function/Data display is not lit, make sure there is power available to the system.
4. Press **Up** or **Down** until 02 appears in the Function/Data display.
5. Press Enter on the control panel.
6. Press Up or Down until B M appears in the Function/Data display. The Function/Data display should show 02 B.
7. Press Enter on the control panel.
8. Press **Power** on the control panel.

The system takes approximately 10 to 45 minutes to power on and progress through an IPL far enough to continue with these instructions. You should see that the data changes in the Function/Data display. The last step of the IPL may take up to 30 minutes to complete or the **Attention** light may turn on.

9. Reference code x6004031 or x6004508 (where x can be any letter) will appear in the Function/Data display and remain for up to 30 minutes.
10. When the system has completed the initial phase of the manual IPL it should show 01 B and you have a console.

Note: Some system reference codes (SRCs) can be displayed without the attention light being on. One example is x6xx450x (where the x can be any letter or number). These SRC codes typically indicate that the system detected an unexpected condition, and the console might have data indicating this condition. This condition and the resulting console data precede the IPL or Install the System window.

If the **Attention** light is lit, then go to step 11.

If the **Attention** light is not lit and you do not have a console, then, consider the following:

- Your system may not have progressed through an IPL far enough to continue with these instructions. Wait at least 30 minutes before going any further.
 - If, after 30 minutes, you do not see any system activity and the Attention light did not light: See the information about handling and reporting system problems in the Troubleshooting and service topic.
 - When the problem has been resolved, start at the beginning of this section again.
11. If you see System Reference Code (SRC) x6xx500x (where the x can be any letter or number) in the Function/Data display, then, go to the Troubleshoot system reference code (SRC) data topic for details.

If you do not see System Reference Code (SRC) x6xx500x (where the x can be any letter or number) in the Function/Data display, then, see the information about handling and reporting system problems in the Troubleshooting and service topic.

Note: If you are working with a console related problem the most common SRC code reported would be A6005008. If you see any reference code of A600500x (where x can be any number) then you can proceed with any console service functions to debug or make changes.

Related reference

[Troubleshooting system reference code data](#)

If you receive any of these system reference codes (SRCs), you might have some problems in your Operations Console configuration.

Using the operations console service functions

Operations console service functions are the emergency operations console recovery functions. This section gives an overview of the functions and details on how the operations console service function works.

Note: This is a service function that requires familiarity of control panel functions. Improper use of this service function can cause the system to become unstable. Call your local support for assistance if you have concerns about making your system unstable.

The operations console service functions are accessed by operator panel functions (65+21). The combination of panel operator functions 65 and 21 is needed to sequence or progress through the service functions (options or commands). The following operations console service functions are available:

- Change the operations console type value **(03-04)**

You can use the operations console service functions to change the operations console type from its current value to another. Two types of operations console are supported: LAN-attached and HMC-attached. For example, you were attached to an HMC (console type 04) and now need to detach to set back to LAN operations console. Use value 03 to change to LAN operations console.

- Display current IP addresses that are used by LAN operations console **(A1)**

This option requires selecting functions 12 and 13 to see the 8 SRC words that represent the IPv4 and IPv6 addresses. Multiple IP addresses might exist, so more operator panel functions (65+21+11+12+13) are required to see any additional addresses.

- Clear the resource and configuration for the LAN adapter that is used by Operations Console **(C3)**

With this option, you can disassociate the current LAN adapter that is used for Operations Console. You might use this option to overcome a mistake in the configuration. For example, assume that you made a typing error and entered another device's IP address. At connection time, the client configured the system's LAN adapter for use by the console, but the console fails to connect because the other device is active. This option clears the system's network data for the console and allows you to delete the client's configuration so that you can start over and make the BOOTP work again.

Depending on your intent to clear the LAN adapter configuration you might also want to stop and restart the LAN adapter. The example here would benefit from following the clear function with a deactivate and activate (A3) function to save time from having to do an IPL.

- Set the server configuration IP address to a known address (192.168.2.150) **(C4)**

This option is used when DHCP or BOOTP fails to set the server IP address.

This option requires the configuration to be cleared first with function C3. Option C3 must be performed with a function 21. Then immediately perform a function 65+21.

Option C4 is displayed. Then use function 21 to perform the set function. The A3 option is required after the C4 option. Before using option A3, you might want to set other configuration options using these options (E1,E2,D1,D2).

- Clear LIC service tool server DCM objects **(C5)**

This option clears the user defined service tool server certificate and SSL control settings. Use this option if the user defined service tool service certificate is not working with the ACS client and the certificate cannot be removed by using the DCM interface. After the information is cleared, a new SSL environment is created and the old environment is closed. The new environment uses the system default certificate and control settings.

After using this option the configuration in DCM will not match with what the service tool server is now using. Once the system is at the command line and the DCM interface is available, synchronize the configuration by removing all user-defined service tool server certificates that are configured for the IBM i System Service application definition.

Using this option does not prevent the DCM objects from being put into effect the next time the DCM configuration is changed.

- Deactivate followed by an activate of the LAN adapter that is used by Operations Console **(A3)**

With this option, you can reset the LAN adapter that is used by Operations Console, when some network problem causes the system to get into a bad state and the console cannot become active. This forces the LAN adapter to deactivate, and then start back up again. This might clear up the problem, providing the original problem that caused the connection failure has been solved.

This option may be used in place of an IPL for some circumstances, such as after a clear of the LAN adapter configuration.

- Dump Operations Console flight recorders to vlogs **(DD)**

Note: This option will not work if the system performs IPL in D-mode.

This option allows you to capture valuable debug information regarding a console connection failure for support personnel. This method is less intrusive than performing a main storage dump which would force an IPL. When used, an attempt is made to gather all the flight recorder logs from the many parts of the code that is used by Operations Console. A set of vlogs is created for major code 4A00 and minor code 0500. These vlogs can then be sent to your service provider for analysis.

Note: When possible, perform an IPL on the system to guarantee all vlogs get created even if the IPL fails. The intent is that LIC has started the vlog tasks prior to performing the dump of flight recorders.

The following functions are available when not managed by an HMC. The values that are displayed depend on the model and code level.

- Enabling or disabling the embedded Ethernet port and add-on LAN adapters **(E1, E2, D1, D2)**

Note: For a list of supported adapters, go to the [Operations Console hardware requirements](#) topic. All the adapters in the table are available for add-on LAN adapter. The HEA is controlled by embedded Ethernet port value and VLAN is controlled by the management console.

- Selecting an individual LAN adapter location (B1, B2, B3, B4, B5, B6, B7, B8, B9, BA, BB, BC)

How operation console service functions work

The section is an overview of how console service functions work. Console service functions allow you to manually step through the console functions to configure the console and restart the adapter for the console. You will cycle through a series of selecting function 65 and function 21 (essentially a paired set of functions) to step through the console functions. Access to the console service functions requires the operator panel be in manual mode. Set the manual mode by completing the following steps:

1. Locate the server's control panel.
2. Press the **Up** Arrow key until you see 02, and press **Enter**.
3. Press **Enter** again. A < (less than symbol) appears next to N. N indicates Normal mode.
4. Press the **Up** Arrow key. The N changes to an M. M indicates Manual mode.
5. Press **Enter**.
6. Press **Enter** twice. A 02 is displayed on the control panel.

Note: In the following steps, when selecting a function with Up or Down, use the arrow on the operator panel.

If you cycle through panel functions and do not see function 65, you must first enable the extended panel functions. Use the following steps to enable the extended panel functions:

1. If not already in manual mode, use function 02 as described above.
2. Using **Up** and **Down**, select function 25. Press **Enter**. The panel responds with 00.
3. Use **Up** to select function 26. Press **Enter**. The panel should respond with 00. If FF is shown, repeat steps 2 and 3.

Perform the first pair of function 65 and function 21.

1. Use **Up** to select function 65. Press **Enter**. The panel responds with 00.
2. Optional: Use **UP** to select function 11. Press **Enter**. SRC D1008065 will be displayed.
3. Use **Up** to select function 21. Press **Enter**. The panel responds with 00.
4. Use **Down** to select function 11. Press **Enter**. SRC A6nn500A will be displayed, which indicates the current console type.

Now that you have done your first paired set of function 65+21, subsequent entering of function 65+21 will result in SRCs being displayed in the format of SRC A6nn500x . Use **Down** to select function 11 for each function 65+21 pair. The values for **x** in the SRC mean:

A6nn500A

You are displaying the current console type setting.

A6nn500B

You did a second function 65+21 pair, so you are in edit mode. From here, additional function 65+21 pairs will cycle through the available console service function options that are identified by the nn in the SRC. For more information on the various options, see [“Console service function options”](#) on page 93. When you reach the one you want, you will select function 21 to perform the displayed function.

A6nn500C

You had used function 65+21 pairs to cycle through the available console service functions and found the function you wanted. You then selected function 21 again to perform the displayed service function nn. For example, you selected function 21 again when SRC A6C3500B was displayed, which caused service function C3 to be performed, which cleared the LAN configuration and then displayed A6C3500C to indicate the C3 service function had completed.

A6nn500D

More than 60 seconds have passed after the last function 65+21 pair. This is the normal way to exit the console service functions. If you had intended to continue make changes, you need to enter edit mode again by selecting function 65+21 pairs.

Console service function options

When you are in edit mode (SRC A6nn500B), you will perform additional function 65+21 pairs, and the service function options will be displayed in the order shown in the list below, where the *nn* in the SRC is identifying the service function option. If you want to do a particular function, you select function 21 again to perform the operation. The options are:

- **0y** = Current console setting.

Since the console cannot be changed back to 0 after it has been one of the other values, and it is not necessary to ever set the console type to 4 because a Hardware Management Console (HMC) would automatically connect and set the console type to 4, the only value to ever set would be 3 to configure a LAN Operations console. If the console type is 04, you must cycle thru all the options until A603500B. The *y* in the current console setting has these meanings:

- **0** = No console configured (factory default setting that indicates the system has never had a console)
 - The next 65+21 pair will be setting the console to LAN.
- **3** = LAN Operations Console
 - Select Function 21 again and press **Enter** if you need to change the console type to LAN.
 - Select Function 11 to confirm that the SRC A603500C is displayed.
- **4** = Hardware Management Console (HMC)
 - Select Function 21 again and press **Enter** if you want to change the console type.
 - Select Function 11 to confirm that the SRC A604500C is displayed.

- **A1** = Display current IP addresses that are used by LAN console.

If you want to see the IP addresses, you will need to select function 12 and press **Enter** and select function 13 and press **Enter**. Note that as you step through pairs of 65+21, A1 will appear once for each unique IP address. IBM automatically generates IP addresses to simplify set-up. As an example, typically the system automatically generates 1 IPv4 and 2 IPv6 IP addresses, so it would look like the following:

- Select function 11 and press **Enter** to display the SRC A6A1500B.
- Select function 12 and press **Enter** to display the next words of the SRC. The display will show 0D200060 A9FE3E27, which is the auto generated IPv4 169.254.62.39; and FE800000 00000000, which is the first half of the auto generated IPv6.
- Select function 13 and press **Enter** to view the remaining words. The display will show 34C06BFF FE6A3C08, which is the second half of the auto generated IPv6 address FE80::34C0:6BFF:FE6A:3C08.
- Select function 65 and press **Enter**. It will display the next set of IP addresses. Note that function 21 is optional.
- Select function 11 and press **Enter**, SRC A6A1500B is displayed.
- Select function 12 and press **Enter**, this will display the next SRC words. The display will show 0D200060 00000000, which indicates no additional IPv4 addresses, and 20020905 150E0501, which is the first half of the next auto generated IPv6 address.
- Select function 13 to get the remaining words. The display will show 34C06BFF FE6A3C08, which is the second half of the next auto generated IPv6 2002:0905:150E:0501:34C0:6BFF:FE6A:3C08.
- The next 65 + 21 pair could be another A6A1500B or the next service function option (C3).

- **C3** = Clear LAN configuration.

This option is most often used when cloning or moving a system, or changing hardware associated with the operations console.

- **C4** = Set the server configuration IP address to a known address (192.168.2.150).

This option is skipped if C3 not performed first. The option is used in the rare cases when the system fails to find the console, and you want another address to try. The A3 option is required after the C4 option to activate the new IP address. Before using option A3, you might want to set other configuration parameters using options E1,E2,D1,D2, and Bn.

- **C5** = Clear LIC service tool server DCM (Digital Certificate Manager) objects.

This option is used in the rare cases when you cannot connect the operations console due to certificate errors or expiration. The certificates are cleared, and a default imbedded certificate will be used.

- **A3** = Deactivate followed by an activate of the Operations Console.

This option resets the console and should be done after all C4, E1, E2, D1, and D2 operations console configuration changes have been made.

- **DD** = Dump all console-related flight recorder into a set of LIC logs.

This option should only be used when directed by IBM support.

- **E1** = Enable Ethernet embedded port.

This option is used to override automatic system configuration. Best practice is to use option E1 or E2, but not both.

- **E2** = Enable add-on LAN adapters.

This option is used to override automatic system configuration. Best practice is to use option E1 or E2, but not both.

- **D1** = Disable Ethernet embedded port.

This option is used to override automatic system configuration. Do not disable both the Ethernet embedded port and the add-on LAN adapters.

- **D2** = Disable add-on LAN adapters.

This option is used to override automatic system configuration. Do not disable both the Ethernet embedded port and the add-on LAN adapters.

- **Bn** = Enable LAN adapter in slot C1, C2, C3, C4, C5, C6, C7, C8, C9, CA, CB, or CC, where *n* is the number of the slot.

Only the slots supported by your server's machine type and model will be shown. Starting with Power10 systems, the slots appear in reverse order to put the most likely choices first.

An example of setting console would be if the console type is 00 (not set) and you want 03 (LAN):

Function	Display	Comments
65	00	Ready to load first option to start access for console service functions. 00 is displayed to indicate success.
11	D1008065	Optionally look to see this initial SRC for loading the first option.
21	00	Perform the operation. In this case, displays the first option in the console service sequence.
11	A600500A	Displays result. In this case, SRC indicates you are in display mode, and the console type is 0 so no console type is set.
65	00	Ready to load next option in the panel function sequence.
21	00	Perform the operation. In this case, moves to the next option in the console service sequence.
11	A603500B	Displays result. In this case, A603500B was loaded and would allow setting of console to type 03 (LAN console).
21	00	Perform the operation. In this case, it sets the console to 03.
11	A603500C	Displays result. In this case, confirms console type is now 03. Repeat Function 11 until this SRC is displayed. This may take several minutes.

You are done at this point, but you may want to see which IP addresses are being used by operations LAN console. The next 65+21 pair and a function 11 will display SRC A6A1500B, where function 12 and 13 can be used to see the IP addresses.

Related reference

[Preparation for your network environment](#)

To prepare for your network environment, you need to identify and comply with the minimum network configuration required to set up an 5250 Console local console on a network (LAN) configuration.

Using the OPSCONSOLE macro

The OPSCONSOLE macro is the system-side debugging and analysis tool for collecting data for or managing console-related work.

IBM-supplied macros are advanced debug and analysis tools resident on the system. These tools are intended to be used only with the direction of support personnel, because inappropriate use of these tools can cause unpredictable problems with your system. If you are not comfortable in the service tools area, you should call your service provider for assistance before using these tools. These instructions assume you do not have a console device but do have another workstation capable of using the system service tools (SST or DST).

Note: Improper use of IBM-supplied macros could result in a change requiring a complete system reload. It is strongly suggested that you use these IBM-supplied macros only at the request of a support representative.

To use the Operations Console IBM-supplied macro support, follow these steps:

1. Access Service Tools using SST or DST.
2. Select **Start a service tool**.
3. Select **Display/Alter/Dump**.
4. Select **Display/Alter storage**.
5. Select **Licensed Internal Code (LIC) data**.
6. Select **Advanced analysis**. (You will have to page down to see this option.)
7. Page down until you find the **OPSCONSOLE** option. Then, place a 1 next to the option and press Enter. You should be on the **Specify Advanced Analysis Options** window. The command should show as **OPSCONSOLE**.
8. Enter the appropriate option and any required parameters in the Options field.
Use the following options based on the function you are running:

- Change the console type = **cnslname x** (where x is 3 or 4)
- Clear the resource and configuration for the Operations Console LAN adapter = **cnfglan -clear**
- Dump Operations Console-related flight recorders to vlogs = **dump -vlog**
- Deactivate the LAN adapter for a local console on a network (LAN) = **deactlan**
- Activate the LAN adapter for a local console on a network (LAN) = **actlan**
- Restart the console device (any console) = **restart**

Note: Use the **restart** option when you need to remove the current console from service and allow the system to determine whether to start or restart a console. You can use the **restart** option to correct a problem with the original console or when switching from one console type to another.

The following are take over Console Information Status window commands:

- To show the Console Information Status window, use the **showstatus** command.
- To not show the Console Information Status window, use the **skipstatus** command.
- To allow the F18 console take over, use the **takeon** command.
- To not allow the F18 console take over, use the **takeoff** command.

The following options are only used when the system is not managed by a Hardware Management Console (HMC):

- Enable the embedded Ethernet port = **enbintlan**. Use this command if you want to use the embedded port for the console.
- Disable the embedded Ethernet port = **disintlan**. Use this command if you want to use a LAN console resource other than the embedded port. You must have enabled support to use the external resource, such as running the macro **enbextlan**.
- Enable an add-on adapter = **enbextlan**. Use this command to take advantage of the faster network capability from one of these adapters instead of your current network adapter.
- Select a specific LAN adapter location = **enblslot x**(x is 1 thru 10 depending on model)
- Clear the LAN adapter location = **clrslot**
- Disable an add-on adapter = **disextlan**. Use this command if you want to prevent the use of this LAN adapter for use as a console.
- Enable both the embedded port and the add-on adapter = **enbboth**. Use this command if you want to make your current model use these resources as the default settings for a new model.
- Disable both the embedded port and the add-on adapter = **disboth**. Use this command if you have a IBM i model and do not want the adapters used for the console.
- Display configuration flags = **dspcfg**. Use this command if a service and support representative requested this information. This macro returns the state of the embedded port, the state of the add-on LAN adapter, the selected LAN adapter slot, and other configuration settings.

Changing the value of the display console status screen option

You can use System Service Tools or Dedicated Service Tools to change the display console status screen.

Using System Service Tools to change the display console status screen

1. Access **System Service Tools (SST)**.
2. Select **Work with service tools user IDs and Devices**.
3. Select **Work with service tools security options**.
4. Type 1 or 2 on the **Display console status screen** input field and press Enter. When set to 1 (Yes), the system will display the console recovery status screen. When set to 2 (No), the system will not display the recovery status screen.

Using Dedicated Service Tools to change the display console status screen

1. Access **Dedicated Service Tools (DST)**.
2. Select **Work with DST environment**.
3. Select **Service tools security data**.
4. Select 13 (Display console status screen) and press Enter. The status can be **Show** or **Skipped**.

Changing the value of the Console F18 takeover option

You can use System Service Tools or Dedicated Service Tools to change the Console F18 takeover.

Using System Service Tools to change the Console F18 takeover

1. Access **System Service Tools (SST)**.
2. Select **Work with service tools user IDs and Devices**.
3. Select **Work with service tools security options**.
4. Type 1 or 2 on the **Allow console device F18 take over** input field and press Enter. When set to 1 (Yes), the system will allow the PF key 18 on the console device. When set to 2 (No), the system will not allow the PF key 18.

Using Dedicated Service Tools to change the Console F18 takeover

1. Access **Dedicated Service Tools (DST)**.
2. Select **Work with DST environment**.
3. Select **Service tools security data**.
4. Select 14 (Console F18 takeover) and press Enter. The status can be **Enabled** or **Disabled**.

Configuring a service host name (interface name)

The service host name (interface name) is the name that identifies the IBM i service connection on your network that is used for service tools, which includes an 5250 Console local console on a network (LAN) configuration.

You need a service host name (interface name) any time a console or remote control panel is being connected using a network connection. One reason to add this function is when a system has been logically partitioned.

Regardless of which method you use to implement the configuration data, the actual name and associated address used for the service host name depends on the network environment in which the system will be placed. The implementation method has no bearing on whether the configured connection is the first connection. Follow these guidelines when entering the name for the service host:

- For small network infrastructures in which only a few devices are connected, you can typically specify anything you want for the name and associated address. When you set up a small network, you can specify the name and an address range.
- For large network infrastructures managed by IT personnel, a specific name might be required. This avoids confusion with other devices on the same network, and might be used to allow the network infrastructure's equipment to know in advance what the connection name is and what address the connection will use to communicate on the network. Alternatively, you might be able to specify an original name, but the address might be given to you by the network administrator.

There are two methods to create a service host name (interface name):

- A service host name (interface name) cannot be created during the manufacturing process for a system that has 5250 Console local console on a network (LAN) configuration ordered. The LAN adapter is installed and the correct console type is specified. Then, when the user gets the system, the 5250 Console configuration wizard, when completed, supplies the system with the customer network parameters, including the service host name (interface name). During the initial connection, this data finishes the system configuration for the network. This process is also known as *BOOTP*.
- The second method to create a service host name (interface name) is by using an existing console. This method could be used during a migration or an upgrade before disconnecting your old console. When using the following procedure, you can either verify or create the configuration for the IBM i service connection. You can find the service host name (interface name) by going into Dedicated Service Tools (DST), or System Service Tools (SST) on the partition that you are configuring and use the Configure Service Tools Adapter display. Enter the same name on the PC as the existing service host name (interface name) defined in DST or SST.

Note: You can also use any option pertaining to the service tools LAN adapter to verify the service host name or data.

To create a service host name (interface name):

Note: To perform the following procedure using SST, select the option **Work with service tools user IDs and Devices** everywhere it says select **Work with DST environment**. Also note that if you are using a console type other than 5250 Console (LAN), you can use the option **Configure service tools LAN adapter** to create or change the service host name or its data.

Note: You might receive a message "No valid LAN adapter available". This message means that no adapter is available or no adapter is tagged for LAN console. Press F13 to obtain a list of LAN adapters that are not in use by the operating system. Place a 1 next to the adapter and press enter, then continue with step 7.

1. Access Dedicated Service Tools (DST) or System Service Tools (SST).

2. Select **Work with DST environment**.
3. Select **System devices** (skip this step if using SST).
4. Select **Select Console**.
5. Select **5250 Console (LAN)** and press Enter. This should show Verify 5250 Console Adapters.
6. Press F11 to configure.
7. The service host name (interface name) field contains the name. If you are creating a new service connection follow these steps:
 - a) Enter the network data in the appropriate fields.
 - b) Store your configuration by pressing F7.
 - c) Activate the LAN adapter by pressing F14.
 - d) Press F3 to exit.

Related reference

[Preparation for your network environment](#)

To prepare for your network environment, you need to identify and comply with the minimum network configuration required to set up an 5250 Console local console on a network (LAN) configuration.

Related information

[Service tools user IDs and passwords](#)

[Access service tools](#)

Deallocating or moving the LAN adapter card from use by Operations Console

During a migration, you might need to deallocate the LAN card from use by Operations Console. You need to deallocate the LAN card if you are not planning on using an Operations Console local console on a network configuration or the service tools server.

After the LAN card is deallocated, you can move it or use it for another purpose. You must also be using a console type other than an Operations Console local console on a network (LAN) or the following steps will cause the console to disconnect. Follow these steps to deallocate the LAN adapter currently associated with an Operations Console local console on a network (LAN):

Note: To perform the following procedure using SST, select the option **Work with service tools user IDs and Devices** everywhere it says select **Work with DST environment**.

1. Access Dedicated Service Tools (DST) or System Service Tools (SST).
2. Select **Work with DST environment**.
3. Select **System devices** (skip this step if using SST).
4. Select **Select Console**.
5. Select **Operations Console (LAN)**. The LAN adapter currently in use need to be shown.
6. Press F11.
7. Press F6 to perform a clear.
8. Press F7 to store the new values.
9. If you are not using this resource for the console, press F13 to deallocate the adapter. You will be required to use another console type or resource on the next IPL.

Note: After you exit this window, do not enter the configuration again. Entering the configuration again will reallocate the LAN adapter resource to Operations Console.
10. Press F12 twice to exit this window. You should have returned to the **Work with System Devices** window. If you are using SST, this returns you to Work With Service Tools User IDs and Devices.
11. Select **Select Console**.
12. Select the console type you want to use.

Important: You must change the console type to something other than Operations Console (LAN) or the adapter will get reallocated on the next IPL.

Related reference

[Changing from one console type to another](#)

Depending on how your console connection is set up, you can change to a different console type.

Related information

[Access service tools](#)

Changing network values for Operations Console (LAN)

If you need to make a change to the network adapter used for Operations Console (LAN), such as a new IP address, use these instructions.

Note: To perform the following procedure using SST, select the option **Work with service tools user IDs and Devices** everywhere it says select **Work with DST environment**.

1. Access Dedicated Service Tools (DST) or System Service Tools (SST).
2. Select **Work with DST environment**.
3. Select **System Devices** (skip this step if using SST).
4. Select **Select Console**.
5. Select **Operations Console (LAN)**. The LAN adapter currently in use should be shown.
6. Press F11.
7. Use one of the following methods to make your change:
 - If you are making a simple change, such as the IP address, enter in the new values and continue with step 8.
 - If you will be changing the adapter card press F6 to perform a clear. Continue with step 8.
8. Press F7 to store the new values.
9. Press F3 until the DST main menu appears.

Important: If the change did not affect the network IP address or the service host name (interface name) you can exit these instructions now.

If you made a change that caused the network IP address or service host name (interface name) to be different for the currently configured connections, this change must be reflected on all PCs that connect to this service host name (interface name). Since you cannot modify the network IP address or service host name (interface name) of an existing connection's configuration on the client you will have to delete the current connection and re-create a new connection using the new network IP address. Continue with the next step.

10. There are two methods for completing the necessary work in order to allow a new IP address or service host name (interface name). The first is using an IPL. This is the recommended method because you will have more control over when you do the remaining work on the PC. The system will continue to use the old values until an IPL or manual intervention. The second method is to perform the manual intervention, at this time. Perform one of the sets of steps below to complete the network changes.

- **Using an IPL**

This method requires that the client reconfiguration be complete prior to establishing the next connection using Operations Console on a network. If you are currently using the console connected via LAN you would normally start an IPL, it is recommended that the IPL be an attended IPL, and you can reconfigure the client during the initial stages of the IPL. You could, for example, use a different PC as the console instead of the one you currently have connected. You could do the configuration on that PC using the steps here, then after the IPL has been started you could disconnect the current console PC's connection and start a connection on the other PC with the newly created configuration. In this manner you could reconfigure the existing client at your leisure, before the next connection to the system.

- a) Start an attended IPL on the system.
- b) Continue with completing the PC changes.

- **Perform the manual intervention**

Perform these steps from the DST or SST main menu.

Note:

- To perform the following procedure using SST, select the option **Work with service tools user IDs and Devices** everywhere it says select **Work with DST environment** and skip the step **Select System Devices**.
- Select **Work with DST environment**.
- Select **System Devices** (skip this step if using SST).
- Select **Select Console**.
- Select **Operations Console (LAN)**. The LAN adapter currently in use should be shown.
- Press F11.
- Press F17 to deactivate and reactivate the LAN adapter card.

Note: This will cause all LAN connected console PCs to go to `Connecting console` as a status. Also, if more than one LAN connected console PC is connected, the selection of the next console device is unpredictable.

Related information

[Access service tools](#)

Troubleshooting Operations Console connection

Problems can occur during an Operations Console session. The following topics are some solutions to common problems that are encountered during your initial setup and management of your configurations.

The Operations Console client (PC) keeps a set of logs that can be used by IBM development in assisting with customer issues. These logs are not meant to be used by the end user. Details of what to capture and its format change often to keep current with the types of issues being reported. If you report a problem to your service provider, you might be asked to send specific logs from the PC, as well as data captured by the system.

Troubleshooting connection problems

When setting up your initial connection, you might encounter problems connecting your Operations Console configuration.

Local console connection problems

When setting up your local console, you might encounter problems connecting. Failure to connect is defined as problems resulting in the status not going to **Connected** and the emulator did not start.

Some connection problems might require the use of the console service functions to make changes to the system or to do a specific function before a connection can be made.

Network connection errors

These are solutions to problems that occur when a local console fails to connect to a system over a network.

Try these possible solutions:

- Make sure the network is working.
- Verify that you provide the correct service tools user ID and password.
- If you are using Ethernet for your network, you can use a cable to directly connect the PC to the adapter temporarily. This cable isolates the PC and system from any potential problems on your network that might interfere with proper operations.

Troubleshooting system reference code data

If you receive any of these system reference codes (SRCs), you might have some problems in your Operations Console configuration.

Refer to the sections below to find information about specific system reference codes and troubleshooting options.

Console-related system reference codes might require the use of the console service functions to change a setting or to do a function.

System reference code A6nn500x

These are the system reference codes (SRCs) used to access console types and console tasks.

These SRCs are associated with the operation of the control panel method to change the console type or accomplish a console task when the console or other workstation is not available.

Remember: nn can be any alphanumeric designation.

- A6nn 500A - You are displaying the current console type setting.
- A6nn 500B - You did a second 65+21 so you are in edit mode.
- A6nn 500C - You executed a second 21 to cause an action, such as setting the console to another value.
- A6nn 500D - You waited too long after entering edit mode to cause an action so you will have to reenter edit mode again if you intended to make the change. A 21 at this time will force the console to DST, not cause an action.

Related reference

[Using the operations console service functions](#)

Operations console service functions are the emergency operations console recovery functions. This section gives an overview of the functions and details on how the operations console service function works.

System reference code A6005001, A6005004, and A6005007

These system reference codes (SRCs) might be displayed for twinaxial consoles.

A6005001

A console resource (controller) was not found during a manual IPL.

A6005004

A console device was not found during a manual IPL. A twinaxial controller was found but may not be used. This is only indicating the presence of a controller. It is not indicating the controller may be defective.

A6005007

A console device was not found during a manual IPL. This SRC is also indicating the presence of hardware that might indicate another console type other than twinaxial was found. This reference code is not indicating a failure of that hardware or that it is the intended console.

These SRCs, as well as the attention light, are reset when a console is detected and becomes active. If one of these SRCs exists for a long time, you might need to perform an IPL to try to find a console device, depending on many factors, including the model and the hardware present. You can force the system to try finding the console again by using function 21 from the control panel, remote control panel, or virtual control panel. You can also use the console service functions to gather data or attempt recovery.

System reference code A6005008

Use these tables if you received system reference code (SRC) A6005008. If an IPL did not find a console and if the console type is set to anything except a 1, the system will display code A6005008.

Keep the following in mind when examining the information in the tables:

- If you just replaced the LAN adapter associated with Operations Console (LAN), you need to wait at least 35 minutes for the system to find and use the new LAN adapter. In this case, when the system is satisfied, it starts using the new adapter. The console should start, and the SRC is no longer shown.
- If you are attempting to use Operations Console, see the information relating to words 13, 14, and 15 since the local console on a network uses these words.

If Word 13 value (yy) is:	Failure	Word 14 means:	Word 15 means:
00	The LAN connection is active and has received TCP/UDP/RAW packets (e.g. ping). However, the PC failed to connect.	IP address	
01	No supported HW detected or HW detected is not expected (for example, you replaced the LAN IOA so the serial number is different)		In some cases the serial number of the expected adapter may be shown.
02	LAN IOA failed to report		
03	Hardware error	Common error codes: <ul style="list-style-type: none"> • 53001A80, 53002AC0, CC10031A: Network, cable or the LAN adapter may not be operational. • 00000000: This error code indicates the adapter reported but has not been initialized yet. This is not considered an error, at this time. The adapter should be activated shortly. • For other error codes contact your service provider. 	Card position or serial number of adapter
04	BOOTP status: If attempts are zero, then BOOTP is ready, when called. If attempts have a value, then the PC did not respond	Attempts	Card position or serial number of adapter

Table 4. Word 13 to 16 mappings (LAN) (continued)

If Word 13 value (yy) is:	Failure	Word 14 means:	Word 15 means:
05	The system's LAN connection is active but the PC failed to connect. Are the PC and system on the same network and using the same protocol? Can the PC ping the system? (ping serverhostname)	IP address	Card position or serial number of adapter
06	LAN is active but IPv6 autoconfiguration has failed	IPv4	IPv6 status
07	LAN is active but no IPv4 address has been obtained through BOOTP and IPv6 autoconfiguration has failed	ssss bbbb, where: <ul style="list-style-type: none"> • ssss = IPv6 status • bbbb = Bootp attempts 	Card position or serial number of adapter
FF	No console resource or port was found		
Word 16		Word 16 in the form of xxxx yy zz means the following: <ul style="list-style-type: none"> • xxxx where: <ul style="list-style-type: none"> – The twinaxial-related SRC is represented by the first 4 characters. • yy where: <ul style="list-style-type: none"> – 0A = No console tagged • zz where: <ul style="list-style-type: none"> – 00 = Not defined by user (old default value) – 03 = Operations Console (LAN) – 04 = Hardware Management Console (HMC) 	

For Word 13 values 00 to 05, the information in [Table 5 on page 103](#) is applicable.

Table 5. Word mappings for Word 13 values 00 to 05 (cable)

If Word 17 value is:	Failure	Word 18 means:	Word 19 means:
1	Async card not detected		
2	No cables detected	Card position	Card type
3	Wrong cable detected	Card position	Cable ID
4	Port in use	Card position	Card type
FA	Not configured for direct cable		

For Word 13 values 06 and 07, the information in [Table 6 on page 104](#) is applicable.

<i>Table 6. Word mappings for Word 13 values 06 and 07 (IPv6)</i>			
If Word 13 value (yy) is:	Word 17 means:	Word 18 means:	Word 19 means:
06		Upper IPv6 interface ID	Lower IPv6 interface ID
07		Upper IPv6 interface ID	Lower IPv6 interface ID

Notes:

- It is expected that a D-mode IPL with a new load source direct access storage device (DASD) will show the console type value of 00. Examples of when this can occur might be that the copy of data from a failing DASD did not copy all data or you are installing a new logical partition. Also, there are times when the DASD is late reporting and the console type value was not retrieved in time. In these cases you can use the console service function to set a console type value or attempt to contact the console again.
- For Word 13 values 06 and 07, Word 15 common IPv6 status codes include the following:
 - 9043 - Duplicate IPv6 address
 - 9046 - Duplicate auto-configured, link-local IPv6 address

System reference code A6005082

Here are some possible reasons you received system reference code (SRC) A6005082.

- If the system displays this SRC, it typically means that a console was found but lost the console connection.
- If the console is reassigned and the system can locate a console, the SRC is no longer shown.
- The console type does not affect this SRC.
- Only issued in attended mode IPLs.

System reference code A9002000

Here are some possible reasons you received system reference code (SRC) A9002000.

- If the system displays this SRC, it typically means that a console was not found by the IBM i operating system.
- The system value QAUTOCFG must be set to **ON**. IBM i is unable to create the new console device if it is off.
- If you just migrated the console from one type to another and the new console fails to work in IBM i you may need to use another workstation to manually delete the controller and device description associated with the old console device.

Note: You might be able to use the console service functions to assist in a recovery or to gather debug data.

IPL step C6004031 takes longer than expected

The IBM i operating system can detect the hardware resource for a console.

Depending on what other console-capable resources might be found plus the time it takes to walk the bus, this activity has increased the amount of time spent at this IPL step in the startup process.

Troubleshooting other Operations Console problems

Here are some problems with your Operations Console not covered in the other troubleshooting sections.

System requests do not work

Here are the solutions for when system requests do not work.

When using Operations Console, **SYSREQ** corresponds to **Shift+ESC** and is defined as the default for the Access Client Solutions 5250 Emulator.

Most keyboards have a **Print Screen** key, also labeled as **SYSREQ** and would be activated by using that key with the **Ctrl** key, however Windows reserves this key for the **Print Screen** function.

You must remap the keyboard by using the operating system, not the Access Client Solutions 5250 Emulator, in order to change it.

Unable to sign on because of a lost or expired password or disabled user ID

You can use this information to correct a problem when the takeover function isn't working.

If you have the special DST sign-on screen but find yourself unable to sign on because of either a disabled user ID or expired password, you can attempt the first steps of recovery by doing the following:

1. Make certain that no other devices (PCs) that normally are eligible to become the console are connected.
2. Perform the console service functions using 65, 21, 21.

This will cause the console to be lost temporarily. The device should then become the console with a sign-on screen appropriate to the system state, assuming that it matches the current console type setting. For example, if the system was IPLed to command entry, then you would see the IBM i sign-on screen. You can then sign on using any user ID with the authority to continue the recovery of the DST user ID that has the problem.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Programming interface information

This IBM i Access publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Product Number: 5770-XJ1