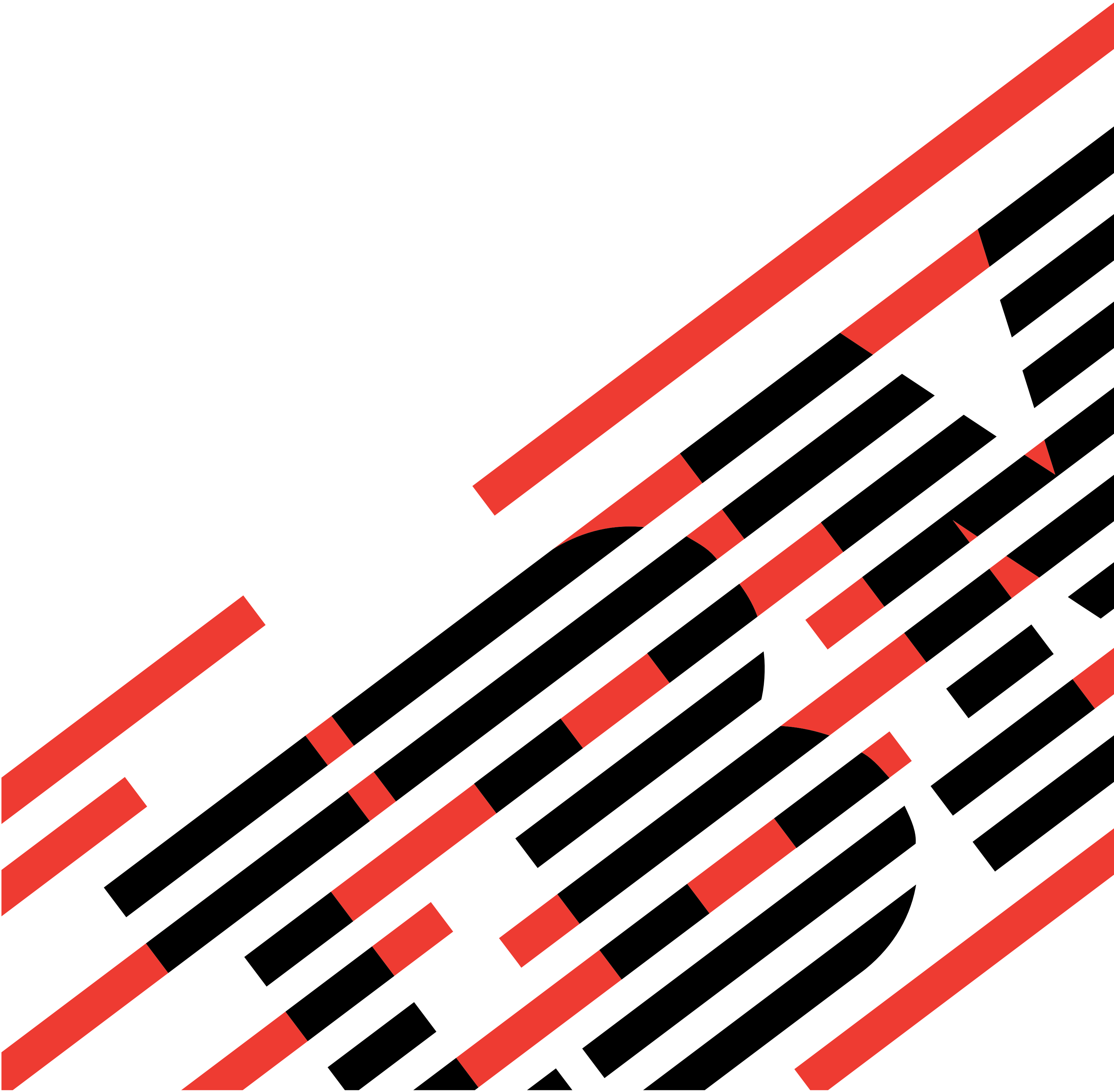
IBM Systems - iSeries

iSeries Access for Windows: Administration

Version 5 Release 4



IBM Systems - iSeries

iSeries Access for Windows: Administration

Version 5 Release 4

Note

Before using this information and the product it supports, read the information in "Notices," on page 149.

Seventh Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of iSeries Access for Windows 5722-XE1 and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

iSeries Access for Windows: Administration

Use this topic to administer iSeries™ Access for Windows® in your client/server environment.

This information assumes that you are familiar with iSeries Access for Windows, and have installed it on your system.

Choose from the following administration topics for additional, required iSeries Access for Windows information:

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 148.

Related concepts

Introduction to iSeries Access for Windows

Programming for iSeries Access for Windows

Related tasks

Installation and set up

Related information

"Code license and disclaimer information" on page 148

What's new for V5R4

Find a summary of the new administration functions for this release.

With the installation of V5R4 iSeries Access for Windows, you can manage your environment through new functions that have been added to the database providers and other product enhancements.

New features for the iSeries Access for Windows administrator include:

- **Data Transfer features**

  V5R4 Data Transfer now includes support for MS Excel Xml Spreadsheet format, 128 byte column names, PC selection of an independent auxiliary storage pool (IASP), and support for creating and overwriting empty query result sets.
- **PC5250 print and emulation**

  V5R4 iSeries Access for Windows PC5250 comes with integrated support for version 5.8 of Personal Communications 5250. Key enhancements for version 5.8 include printer session association, configuration settings in the session windows in a printer session, page and printer setup APIs, additional Bidi enablement, and automation object support for .NET.
- **iSeries Navigator**

  There are several new iSeries Navigator features. For a description of these features see information provided with iSeries Navigator.
- **ODBC**

  In V5R4, ODBC and your iSeries host support 128-byte column names and longer SQL statements (commands up to 2,097,152 bytes or 1,048,576 characters). ODBC also supports passing an IBM® Enterprise Workload Manager (eWLM) correlator to the iSeries host.
- **Database providers**

  Library List and System Naming are now supported by the **.NET provider** and the **OLE DB provider**. In addition, these providers also support the new V5R4 iSeries host server enhancements, which include 128-byte column names, longer SQL statements, and passing the IBM Enterprise Workload Manager (eWLM) correlator.

The .NET provider also supports LOB data types and customizable String processing for other data types. It also supports multiple active result sets for each connection, and IntelliSense.

  - For technical details about the IBM.Data.DB2.iSeries provider, see the *IBM DB2® UDB for iSeries .NET Provider Technical Reference*. For details about the other providers, see the *OLE DB Technical Reference*. You can access these documents from topics in the *Programmer's Toolkit*, following this path:

    Start → **Programs** → **IBM iSeries Access for Windows** → **Programmer's Toolkit** → **Programmer's Toolkit** → **Common Interfaces**

- **Printer Drivers**

  Beginning with V5R4, iSeries Access for Windows provides a 64-bit AFP™ printer driver for use on 64-bit versions of Windows operating systems. The new driver is supported on the Intel® Itanium (Intel 64-bit) Processor Family of personal computers.

  Notes:

  - The driver is not supported on the Advanced Micro Devices (AMD) Hammer family of processors.
  - The SCS Printer Driver is not supported on 64-bit Windows operating systems.

- **Secure Sockets Layer (SSL)**

  - Beginning with V5R4, you can configure client PCs to, optionally, switch in and out of FIPS-compliant (Federal Information Processing Standards) mode, for most functions of iSeries Access for Windows where SSL is used.
  - Also, beginning with V5R4, Client Encryption (CE3) is no longer installed as a separate product on the server in order to install 128-bit SSL Encryption on your PC. 128-bit SSL Encryption is packaged with the iSeries Access for Windows (XE1) product, and is therefore available as an installable component of your usual new, upgrade, selective, or tailored install options.
  - In addition, starting with V5R4, SSL is available for 64-bit applications on personal computers powered by the Intel Itanium (Intel 64-bit) Processor Family.

    Notes:

    - SSL is not yet available for 64-bit applications run on the Advanced Micro Devices (AMD) Hammer family of processors or on Intel processors with EM64T.
    - SSL is available to 32-bit applications that run on either platform.

Other information

After installing iSeries Access for Windows, use this path from the iSeries Access for Windows folder to access the User's Guide: **Start** → **Programs** → **IBM iSeries Access for Windows** → **User's Guide**.

The C/C++ Database APIs (Optimized SQL APIs) are no longer being enhanced. At some point in the future, support for these may be removed. It is recommended that you use one of the other technologies for database access.

The Windows 98 (all editions), Windows ME, and Windows NT® operating systems are not supported with V5R4 iSeries Access for Windows.

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.

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

Related information

.NET programming

OLE DB programming

What's new for iSeries Navigator in V5R4

Printable PDF

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

To view or download the PDF version of this document, select Administer iSeries Access for Windows (about 436 KB).

Saving PDF files

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

1. Right-click the PDF in your browser (right-click the link above).
2. Click **Save Target As** if you are using Internet Explorer. Click **Save Link As** if you are using Netscape Communicator.
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

Downloading Adobe Acrobat Reader

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

iSeries Access for Windows network environments

Learn how iSeries Access for Windows allows you to make services, that are on your server, available to client PCs, in different network environments. Also learn how to administer a PC that has multiple users.

This topic identifies some of the network environments in which iSeries Access for Windows can operate. You can make i5/OS™ services available to your clients by using iSeries Access for Windows in a three-tier environment, or by installing it on a version of the Windows operating system that provides support for remote logon using Terminal Services. You can administer a PC that has multiple users assigned to it.

Choose from the topics below for information on several methods provided for end users to access to iSeries services using iSeries Access for Windows. Typically this involves a direct connection between a PC running iSeries Access for Windows and the iSeries server. However, using Microsoft® Windows Terminal Server Edition (TSE) environment or iSeries Access for Windows in a three-tier environment allow you to take advantage of other networking environments.

Also choose from the topics below to learn ways provided by iSeries Access for Windows to administer PCs with multiple users:

Microsoft Windows Terminal Server

Use Microsoft Windows Terminal Server features with iSeries Access for Windows.

Microsoft Windows Terminal Server is a feature that allows multiple, simultaneous client sessions to run on a single Windows server. It allows connections from multiple client platforms, including not only

Windows, but network stations, UNIX®, Linux®, DOS, OS/2®, and others. By installing iSeries Access for Windows on a Windows server that provides this feature, workstations that do not have iSeries Access for Windows installed can access iSeries services.

Note: Set **When to check service level** to **Never** on the **Service** tab of iSeries Access for Windows Properties when running Terminal Services and using Windows 2000, and later, operating systems.

For information on installation, support, known problems, and solutions when using iSeries Access for Windows with a Microsoft Windows Terminal Server, refer to APAR II11373.

For more information about Terminal Services on a Windows server, consult Microsoft documentation or their Web site.

Related information

APAR II11373

Windows NT Server 4.0 Terminal Server Edition

Use iSeries Access for Windows in a three-tier environment

By installing iSeries Access for Windows on the middle tier of a three-tier environment, a wide variety of client workstations can access iSeries services.

Additionally, three-tier environments present several other advantages:

- **Improved integration between diverse clients and server applications:** Multiple end-user applications running on various clients can communicate with multiple applications on a Windows server simultaneously. Each of the applications on the Windows server can also, simultaneously, communicate with multiple databases.
- **Enhanced transaction management using Microsoft Transaction Server (MTS):** Three-tier environments allow for more complex transactions, some of which may depend upon each other for their own successful completion. (All transactions must complete successfully in order for any of them to complete.)
- **Importing data from an iSeries server into Web pages, using Microsoft Internet Information Server (IIS):** IIS can use Active Server Pages to dynamically update Web pages with data from a DB2 Universal Database™ for iSeries.

All three-tier environments separate components and applications into three layers. The three layers may reside on separate PCs, or terminals, and communicate over a network. Generally the tiers will have the following characteristics:

Client tier

This layer contains the interface and applications that allow end users to manipulate data. For example, this may involve a Web browser running on a network station, or a custom-built application using a remote component. This layer does not use the iSeries Access for Windows client.

Middle tier

This layer contains the business or application logic. In environments using iSeries Access for Windows, this layer should consist of a Windows server running a Microsoft Active Server Pages script or a remote component.

This layer uses Microsoft's Internet Information Server (IIS) and can, optionally, use Component Services or Microsoft Transaction Server for distributed transactions. The script uses the ADO.NET provder, OLE DB provider, or ODBC driver that are included with iSeries Access for Windows. These clients communicate with the database tier to get data from the iSeries server.

Refer to the following topics for more information about the middle-tier:

- Use Microsoft Transaction Server (MTS)
- Access iSeries services from the middle tier

Database tier

This layer usually consists of a DB2 Universal Database for iSeries database. Your applications can access this and various iSeries services through host server programs, or through custom-built iSeries programs.

Use Distributed transaction support

The iSeries Access for Windows client supports Microsoft Transaction Server (MTS) and the Component Services model, with the iSeries Access ODBC driver and the IBMDASQL OLE DB provider.

MTS

MTS is a Microsoft component-based programming model and run-time environment for developing, deploying, and managing Internet server applications. In many three-tier environments, Active Server Pages (ASP) call MTS components to access databases, mainframe applications, and message queues. Used with iSeries Access for Windows running in the middle-tier of a three-tier environment, MTS components manage transactions between client applications, iSeries Access for Windows components, and the databases involved in the transactions.

MTS uses Microsoft Distributed Transaction Coordinator (MSDTC) in order to manage transactions that span multiple Database Management Systems (DBMS), and to ensure two-phase commit integrity when dealing with transactions whose implementations depend on mutual success.

In newer Windows server models, MTS has been replaced with the Component Services model. The iSeries Access for Windows ODBC and OLE DB providers support the Component Services model in the same manner as they support MTS.

Implementation notes

- If the MSDTC cannot load the iSeries Access ODBC driver, the SQLSetConnectAttr(SQL_ATTR_ENLIST_IN_DTC) will fail with reason code of 2 (XaRmCreate failed). If you installed iSeries Access for Windows PC5250 eumlator component, the MSDTC system environment path is set for you. To avoid this, the system environment path on the PC running MSDTC must include the path to the Shared directory within the directory in which iSeries Access for Windows is installed. For example: C:\Program Files\IBM\Client Access\Shared.
- If you are using SSL, or any other configurable value on the **Connections** → **Properties** dialog in iSeries Navigator, your iSeries connection name in iSeries Navigator must match the connection name specified on the client PC managed by MTS. MSDTC uses the same connection names as iSeries Access for Windows ODBC client PCs managed by MTS to connect to the DB2 UDB for iSeries database. To change the connection properties of the MSDTC connections, you must change the system account registry.

  One way to do this is to use Incoming Remote Command (IRC) in combination with the CWBENV utility:

1. Run CWBENV on a client PC to extract the configuration information for an environment.
2. Copy the resulting file to the MSDTC PC.
3. Start the iSeries Access for Windows Remote Command service and ensure that it is configured to run in the Local System context.
4. Using the RUNRMTCMD command from a PC5250 session, send a CWBENV command to the MSDTC PC to import the environment.

   See the User’s Guide in the iSeries Access for Windows program group for more information on these functions.

For more information about MTS or the Component Services model, refer to the Microsoft Web site.

Related information

Microsoft MTS Web site

Access iSeries services from the middle tier

There are several ways to provide your middle-tier components with access to the iSeries server.

Note: Middle-tier components cannot have a user interface; therefore, if iSeries Access prompts for sign-on information, your three-tier applications may appear to hang. To prevent this, developers must use a new system object to specify required connection information (user ID and password) to the iSeries server. The prompt mode value for this object must be **prompt never**.

iSeries Access for Windows .NET Data Provider

The **IBM DB2 UDB for iSeries .NET Provider** offers the best performance to access the iSeries database for programmers that write applications using Microsoft's .NET Data Access Framework. Throughout this documentation, **Managed Provider** is used interchangeably with **IBM DB2 UDB for iSeries .NET Provider** and **IBM.Data.DB2.iSeries data provider**. Regardless of the name that is referenced, you can take advantage of the full set of .NET data types and SQL functionality to make it easy for applications to work with data stored securely in your iSeries server databases.

See .NET programming for more information.

iSeries Access for Windows OLE DB provider

Most applications and components use the iSeries Access for Windows OLE DB provider through ActiveX Data Objects (ADO). Here are the four primary benefits to implementing this technique:

- It allows your developers to make only minor modifications to a single interface and programming technique in order to access iSeries programs, commands, SQL queries, stored procedures, and physical and logical files.
- It supports automatic data conversions between iSeries and PC data types.
- It allows you to avoid the overhead associated with SQL by providing support for record-level file access.
- It is relatively easy to implement and to develop applications. This method is generally the most simple technology for developing three-tier applications.

See OLE DB programming for more information.

iSeries Access for Windows ODBC driver

Additionally, you can access the iSeries Access ODBC driver through either ADO or Remote Data Services (RDS), by using the Microsoft OLE DB provider for ODBC (MSDASQL).

For more information about accessing ODBC through ADO, see Choosing an interface to access the ODBC driver.

For other iSeries Access ODBC driver information, see ODBC programming.

Note: The iSeries Access for Windows OLE DB provider, and several functions in the iSeries Access ODBC driver, require MDAC version 2.5 or later.

ActiveX automation objects

The iSeries Access for Windows client provides a library of ActiveX automation objects that your developers can use for middle-tier development. These objects provide access to:

- iSeries data queues
- Remote commands and distributed program calls
- Administration objects
- iSeries system objects
- Data Transfer access to iSeries database tables

In some cases, ActiveX objects provide greater versatility and functionality than ADO, but require slightly more complex programming.

Note: The iSeries Access for Windows client includes the automation library from the Windows 95/NT client (the XD1 product). These automation objects, including database, do not support use in a three-tier environment.

Express C/C++ APIs

iSeries Access for Windows APIs provide fast, low-level access to i5/OS host servers. However, using these APIs requires developers who are experienced with C/C++. Specifically, developers must be familiar with C APIs and data types, and must also account for thread-safety considerations when creating their components.

Related tasks

Choosing an interface to access the ODBC driver

Related reference

.NET programming

OLE DB programming

ODBC programming

Add TCP/IP configuration to all users

Use the `CWBCFG` command, from a command prompt or from **Start** → **Run**, to configure iSeries server connections for all users defined on a PC.

Using this command also adds configuration information to the Windows default user profile, which is the profile used when creating additional user profiles.

You can also use `CWBCFG` to add or change the location that the PC5250 emulator uses when it opens or creates files. `CWBCFG` can change the location setting for all users of the PC.

| Finally, you can use `CWBCFG` to turn the FIPS Mode switch on or off for all users of the PC.

| For more information on `CWBCFG` or FIPS Mode, see the online iSeries Access for Windows User's Guide.

Set PC5250 files location for all users

The default location in which PC5250 emulator searches and stores all files for all defined users is shared by all the users of a PC although some may not have authority to write to it.

The default location is:

(iSeries Access for Windows installation folder)\emulator\private

This default location can be changed by each authorized user from the PC5250 tab of iSeries Access for Windows Properties. To change this default location for all users at once, the administrator can use the CWBCFG command from a command prompt, specifying the /pc5250path option.

Notes:

- Any user account created after CWBCFG is run uses the default location set by CWBCFG.
- Only Administrators can use CWBCFG.
- CWBCFG does not move any files from the old to the new location. Files must be moved manually, if desired.

For more information about CWBCFG, see the online iSeries Access for Windows User's Guide.

User profiles for PCs with multiple users

You can administer PCs with multiple iSeries Access for Windows users. This type of administration is available as a function of the Windows operating systems through the use of roaming and mandatory profiles.

Note: For documentation on how to implement these methods of multiple user administration in your network, see Microsoft offerings for the Windows operating system you are using.

Roaming user profiles

The roaming user profiles are Windows user profiles that can roam between PCs. The configuration changes go with the user. The roaming user profiles generally reside on a Windows server. Each roaming user has a directory on the Windows server specified by the user profile path in the user profile settings. This directory contains registry information as well as start menu and desktop information for each user.

Mandatory user profiles

Mandatory user profiles are user profiles that a system administrator sets up for use by PC users on any Windows PC. These users typically should not modify their settings. Mandatory user profiles can exist on one PC or roam between PCs.

ODBC administration

iSeries Access for Windows includes an ODBC driver that can allow your applications convenient access to DB2 UDB for iSeries databases in your network. This topic provides an overview of ODBC, instructions for setting up the driver, and a troubleshooting guide.

Note: For information and considerations when working with the ODBC APIs, refer to ODBC programming.

Open Database Connectivity (ODBC) is a Microsoft standard for providing access to databases. It has a well-defined set of application programming interfaces (APIs) that use Structured Query Language (SQL) to access databases.

For help with integrating ODBC support into your applications, refer to the iSeries Access for Windows ODBC programming, where you can get information on the following subtopics:

- ODBC API list
- ODBC API implementation
- Programming examples
- ODBC performance

Related concepts

iSeries ODBC Driver for Linux
See this topic on installing and using the IBM ODBC Driver for Linux to access the iSeries database. IBM iSeries ODBC Driver for Linux is not part of iSeries Access for Windows. It is a separate product used only with the Linux operating system.

Related reference

ODBC programming

Overview of the iSeries Access ODBC driver

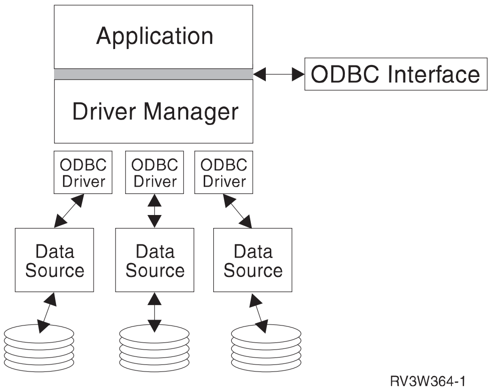
Provides a general description of ODBC, and how you can use it with iSeries Access for Windows.

The iSeries Access ODBC driver is a collection of application programming interfaces (APIs) for accessing database information using Structured Query Language (SQL). Using the iSeries Access ODBC driver allows applications to access different databases on the iSeries server using the same source code, and to handle data in the format most convenient for those applications. ODBC provides an application developer a relatively simple model for creating portable applications or components that must deal with multiple DBMSs.

The ODBC architecture involves an application, driver manager, ODBC driver, and a data source. iSeries Access provides both a 32-bit and 64-bit ODBC driver. The 64-bit ODBC driver is automatically installed along with the 32-bit ODBC driver when running under a 64-bit version of Windows . ODBC applications running in 64-bit versions of Windows will automatically use the appropriate ODBC driver, depending on what bit version the application was compiled for. For example, the 64-bit driver can only be used by a 64-bit application.

In order for an application to use ODBC you must set up a data source. You can use the ODBC Administrator to set up a data source. There are two versions of the ODBC Administrator, 32-bit and 64-bit, that can be accessed from the iSeries Access for Windows folder. When using ODBC Administrator, you have the option to setup three different types of data sources: User, System, and File data sources. For more information about how data sources are configured, see 64-bit ODBC Support, in the iSeries Access for Windows' User's Guide.

ODBC Components



Application. Performs processing and calls ODBC functions to run SQL statements.

Driver manager. Processes ODBC function calls and forwards the requests to the driver.

Driver. Processes ODBC function calls, submits SQL requests to a specific data source, and returns results to the application.

Data source. To use a data source you have to create a Data Source Name (DSN). A DSN contains information about how to access the DBMS. You can specify any of the following DSNs:

- **User DSN:** These data sources are local to a computer, and may only be available to the user who created them. This information is stored in the registry.
- **System DSN:** These data sources are local to a computer, rather than dedicated to a user. The system, or any user having privileges, can use a data source set up with a system DSN. This information is stored in the registry.

Note: On a PC with a 64-bit processor, the system part of the registry is split into 32-bit and 64-bit pieces. System DSNs configured using the 32-bit ODBC Administrator are available only to 32-bit applications. Also, System DSNs configured using the 64-bit ODBC Administrator are available only to 64-bit applications.

- **File DSN:** These are file-based data sources that may be shared between all users that have the same drivers installed so that they have access to the database. These data sources do not need to be dedicated to a user, or to be local to a computer.

For more information about ODBC, refer to the Microsoft Web site.

Related tasks

"Specify the ODBC data source" on page 11
You must specify the data source for your application to access and manipulate data.

Set up your system for the iSeries Access ODBC driver

Presents procedures for setting up your environment to support the ODBC driver. For help configuring the ODBC driver, start the ODBC administration program from the iSeries Access for Windows program group, and refer to the online help.

The iSeries Access ODBC driver is an ODBC version 3.5 compliant driver. The driver requires Microsoft Data Access Components (MDAC) version 1.5 or higher. Applications that use Microsoft ActiveX Data Objects (ADO) should have MDAC version 2.1 or higher installed. The runtimes for MDAC versions 2.1 and later provide additional function for applications that use ADO, the Microsoft OLE DB provider for ODBC, and iSeries Access for Windows ODBC to access their iSeries data. If an application uses connection pooling or Microsoft Transaction Server (MTS) support, it is recommended that the latest MDAC version be installed. You can download MDAC from the following Microsoft Web Site: www.microsoft.com/data.

See the ODBC data source topic to configure your ODBC driver. Complete your configuration by following the steps identified by the topic adding the local system to the RDB directory.

Using independent ASPs through ODBC is optional. See independent ASPs for more information about configuring this support.

For help configuring options for a specific data source, start the ODBC Administrator from the iSeries Access for Windows program group, select the data source to configure, and refer to the online help.

Related information

www.microsoft.com/data

Adding the local system to the RDB directory

To use ODBC, OLE DB, or the .NET Data Provider, the local system name must appear in the RDB directory.

To add the local system to the RDB directory:

1. From the command prompt run the CL command, Add Relational Database Directory Entry (ADDRDBDIRE).
2. When the ADDRDBDIRE screen prompts you for values, enter the name of the system as the Relational Database parameter.
3. Enter *LOCAL as the Remote Location parameter.

There may be additional steps to get the database (RDB) name set, if the version of your system is V5R2 or later and your application accesses data in independent ASPs. The RDB name corresponds with a namespace that consists of the system ASP and any user ASPs or linked ASP group associated with the system ASP. For more information about independent ASPs, see Disk management.

Note: ODBC allows the use of fully qualified names in the format of [catalog name].[schema name].identifier (for example, where identifier is the name of a table, view, or procedure). In the DB2 UDB for iSeries implementation of SQL this corresponds to [RDB name].[collection name].identifier.

Related concepts

Disk management

Specify the ODBC data source

You must specify the data source for your application to access and manipulate data.

To specify the data source:

1. Start the ODBC Administration program from the iSeries Access for Windows program group.
2. Select the appropriate tab for the type of data source. See Overview of the iSeries Access ODBC driver for more information.
3. Select an existing data source from the list, or select **Add** to create new one. If you are using an existing data source, click **Configure** and proceed to step 5.
4. Select the iSeries Access ODBC driver for your data source, and click **Finish**.

   Note: You may notice the Client Access ODBC Driver (32-bit) name in the list of drivers. This name is listed so that data sources created with previous releases of Client Access will continue to work. Both names point you to the same ODBC driver. You can use either name, however in future releases the Client Access ODBC Driver (32-bit) name will be removed.
5. Specify desired options using the iSeries Access for Windows ODBC setup dialog. For a description of the controls, refer to the data source's online help by using the F1 key or the Help button.

Note: The data source name can include up to 32 characters, must start with an alphabetic character, and cannot include the following characters:

Unallowed data-source characters	
Left bracket ([)	Question mark (?)
Right bracket (])	Asterisk (*)
Left brace ({)	Equal sign (=)
Right brace (})	Exclamation point (!)
Left parenthesis ( )	At sign (@)
Right parenthesis ( )	Semicolon (;)

Related concepts

"Overview of the iSeries Access ODBC driver" on page 9
Provides a general description of ODBC, and how you can use it with iSeries Access for Windows.

Disk management

Related tasks

"Use independent ASPs through ODBC"
Find steps to use when connecting to an independent ASP through ODBC.

Use independent ASPs through ODBC

Find steps to use when connecting to an independent ASP through ODBC.

To use **independent ASPs** through ODBC, configure your ODBC DSN and do the following:

1. Select the **Server** tab.
2. Click on ″Override default database with the following:″.
3. Specify the **RDB name** that corresponds with the **Independent ASP** to connect.

4. If no RDB name is specified, the default RDB name is determined from the job description of the user profile that is making the ODBC connection. By default, the driver uses the setting of the user profile for the user making the ODBC connection.

   For more information about **independent ASPs**, see Disk management content topics.

   Related concepts

   Disk management

   Related tasks

   "Specify the ODBC data source" on page 11
   You must specify the data source for your application to access and manipulate data.

iSeries Access for Windows ODBC security

Highlights a few security considerations when working with ODBC, and provides references to more detailed security instructions.

The following information is not intended to be a comprehensive guide to security strategies on the iSeries servers or with iSeries Access for Windows. It simply provides an overview of security strategies that impact iSeries Access for Windows and ODBC users. For more in-depth information, see the IBM Security - Reference.

Related information

IBM Security - Reference

Common ODBC strategies that are not secure

Avoid some common ODBC security techniques to ensure your environment is secure.

Sometimes system administrators attempt to secure access to the data, rather than securing the data itself. This is extremely risky, as it requires that administrators understand ALL of the methods by which users can access data. Some common ODBC security techniques to avoid are:

Command line security

This may be useful for a character-based interface or for 5250 emulation-based applications. However, this method assumes that if you prevent users from entering commands in a 5250 emulation session, they can access data only through the programs and menus that the system administrator provides to them. Therefore, command line security is never truly secure. The use of iSeries Access policies and Application Administration improve security, and use of object level authority improves it even more.

Potentially, iSeries Access for Windows policies can restrict ODBC access to a particular data source that might be read only. Application Administration in iSeries Navigator can prevent ODBC access.

For additional information, see the IBM Security - Reference.

User exit programs

A user exit program allows the system administrator to secure an IBM-supplied host server program. The iSeries Access ODBC driver uses the Database host server: exit points QIBM_QZDA_INIT; QIBM_QZDA_NDBx; and QIBM_QZDA_SQLx. Some ODBC drivers and iSeries Access for Windows data access methods (such as OLE DB) may use other host servers.

Journals

Journaling often is used with client/server applications to provide commitment control. The journals contain detailed information on every update made to a file that is being journaled. The journal information can be formatted and queried to return specific information, including:

- The user profiles that updated the file

- The records that were updated
- The type of update

Journaling also allows user-defined journal entries. When used with a user exit program or trigger, this offers a relatively low-overhead method of maintaining user-defined audits. For further information, see the Backup and Recovery.

Data Source Name (DSN) restrictions

The iSeries Access ODBC driver supports a DSN setting to give read-only access to the database. The iSeries Access ODBC driver supports a read-only and a read-call data source setting. Although not secure, these settings can assist in preventing inadvertent delete and update operations.

Related information

iSeries Security - Reference

Backup and Recovery

ODBC program security strategies

Consider the following ODBC program security strategies.

Restricting program access to the database

System administrators often need to limit access to particular files, to a certain program, or to sets of programs. A programmer using the character-based interface would set restrictions by using program-adopted authority. A similar method can be used with ODBC.

Stored procedures allow ODBC programmers to implement program-adopted authority. The programmer may not want users to be able to manipulate database files by using desktop applications such as Microsoft Access or Lotus® 1-2-3®. Instead, the programmer may want to limit database updates to only the programmer's application. To implement this, user access to the database must be restricted with object-level security or with user exit programs. The application must be written to send data requests to the stored procedure and have the stored procedure update the database.

Restrict CPU utilization by user

ODBC has greatly eased the accessibility of iSeries data. One negative impact has been that users may accidentally create very CPU-intensive queries without realizing it. ODBC runs at an interactive job priority and this can severely affect system performance. The iSeries supports a **query governor**. ODBC can invoke the query governor (for example, through the PC application) in a stored procedure call. Or the ODBC APIs can invoke the governor by way of the query time-out parameter. Also, a user exit program can force the query governor on the ODBC job. The time limit is specified on the QRYTIMLMT parameter of the CHGQRYA CL command. The query options file (QAQQINI) can also be used to set the value.

The *SQL Reference* book contains additional information. View an HTML online version of the book, or print a PDF version, from the DB2 Universal Database for iSeries SQL Reference.

Also see Host server administration for more information.

Audit logs (monitoring security)

Several logs can be used to monitor security. QHST, the History Log, contains messages that relate to security changes that are made to the system. For detailed monitoring of security-related functions, QAUDJRN can be enabled. The *SECURITY value logs the following functions:

- Changes to object authority
- Create, change, delete, display, and restore operations of user profiles

- Changes to object ownership
- Changes to programs (CHGPGM) that adopt the owner's profile
- Changes to system values and network attributes
- Changes to subsystem routing
- When the QSECOFR password is reset to the shipped value by DST
- When the DST security officer password is requested to be defaulted
- Changes to the auditing attribute of an object

For additional information, see the IBM Security - Reference.

Related concepts

"Host server administration" on page 27
Describes the host servers that are commonly used with iSeries Access for Windows, and describes how to effectively manage and use them.

Related reference

DB2 Universal Database for iSeries SQL Reference

Related information

IBM Security - Reference

Related information for ODBC security

Locate additional information on ODBC security.

In-depth security reviews and assistance to implement the strategies above is available through IBM Consultline (1-800-274-0015). Review the following for in-depth information on specific topics:

- Host server administration
- IBM Security - Reference
- Backup and Recovery
- DB2 Universal Database for iSeries SQL Reference
- Go to **Client Access ODBC and OLE DB Security Issues** Technical Reference, which can be accessed by the following instructions:
  - Go to www.ibm.com/servers/eserver/iseries/support
  - **Go to Find it fast! → Search Technical databases**
  - Enter the title (Client Access ODBC and OLE DB Security Issues) as the search criteria.

Troubleshoot ODBC

Helps you solve a few of the more commonly encountered difficulties with iSeries Access for Windows and ODBC. It also identifies several tools that can help you remove performance bottlenecks. You should review this information before contacting technical support.

For help with integrating ODBC support into your applications, refer to the iSeries Access for Windows ODBC programming, where you can get information on the following subtopics:

- ODBC API list
- ODBC API implementation
- Programming examples
- ODBC performance

The following topics provide general guidelines for finding and resolving iSeries Access for Windows ODBC errors:

Related concepts

ODBC programming

ODBC diagnostic and performance tools

Use tools to help diagnose ODBC problems.

Choose from the following for information on ODBC client or server-side diagnostic and performance tools:

Related concepts

"Checking the server status" on page 19
Use CWBPING.

"Gather information for IBM Support" on page 25
The IBM Support staff can offer you better service, if you have certain information available when you open a problem record to IBM Support.

Client-side ODBC diagnostic and performance tools:

Use client-side tools to help diagnose ODBC problems.

The following table contains ODBC client-side diagnostic and performance tools:

ODBC Trace (SQL.LOG)	Microsoft's ODBC Administrator provides its own trace utility to trace ODBC API calls from applications. See Collecting an ODBC Trace (SQL.LOG) for more information.
ODBC trace utilities	There are other ODBC trace utilities available that can be more robust than the ODBC Trace (SQL.LOG). These retail utilities can provide detailed entry and exit point tracing of ODBC API calls. Two tracing utilities are Trace Tools (Dr. DeeBee) and SST Trace Plus (Systems Software Technology).
CWBPING	To use CWBPING, type `cwbping (your system name or IP address)` at a command prompt. For example: `cwbping testsys1` or `cwbping 127.127.127.1` CWBPING responds with a list of servers, and their status. Run CWBPING without any parameters for help with using CWBPING. For more information about CWBPING, see Checking the server status.
CWBCOTRC	To use CWBCOTRC, type **CWBCOTRC ON** at a command prompt while located in the \Program Files\IBM\Client Access directory. After turning on the trace, you can start your application. Typing **CWBCOTRC OFF** stops tracing. CWBCOTRC gathers information about data that is being transmitted to and from the server. Run CWBCOTRC without any parameters for help with using CWBCOTRC.
Detail trace	Detail trace gathers information traced out by the iSeries Access for Windows components that are in use. ODBC information that can be found in this trace includes entry points into the driver, information about the prestart job, the package name in use, and special error conditions. For more information, see Gather a detail trace.

Server-side ODBC diagnostic and performance tools:

Use server-side tools to help diagnose ODBC problems.

The following tables contain ODBC diagnostic and performance tools the server side:

Server-side tools

Communications trace	The communications trace facility will trace and format any communications type that has a line description (token ring and Ethernet). This is a tool for isolating many problems. It also is a useful aid for diagnosing where a performance delay is occurring. Use the timestamp and eye-catcher fields to measure how long it takes to process a request.
Job traces	The job trace can help isolate most host problems and many performance issues. A service job must first be started on the job to be traced. Locate the fully qualified job name of the ODBC job. From any 5250 emulation session, start a service job on this QZDASOINIT job by using the STRSRVJOB command. Then choose one of two traces, depending on the information needed: **Trace job** Traces the internal calls made by the host server. Run the TRCJOB *ON command. **Debug trace** Used to review the performance of your application and to determine the cause of a particular problem. The STRDBG command runs against an active service job. This command logs the decisions made by the query Optimizer to the job log of the debug session. For example, it records estimated query times, access paths used, and cursor errors. An easy way to enable STRDBG is to configure the ODBC DSN you are using through **ODBC Adminstrator** by selecting the **Enable the Start Debug (STRDBG) command** option on the **Diagnostic** tab. Alternatively, you can run the following command: `STRDBG UPDPROD(*YES)` The ODBC job log can record all errors that occur on the iSeries server. When the job is in debug mode, the job log also will contain performance-related information.
Performance tools	Performance toolkit provides reports and utilities that can be used to create an in-depth analysis of your application performance. The toolkit provides information about CPU utilization, disk arm utilization, memory paging and much more. Although the base operating system includes the ability to collect performance data, you will need the separately licensed program **Performance Tools/400** to analyze the results. You can also use the tools Database Monitor and Visual Explain. Refer to the iSeries Navigator Online help for more information.
QZDASOINIT job log	To receive optimal support, generate, locate and retrieve the QZDASOINIT job log. The job log may contain messages that can help you to determine and resolve errors that are returned through ODBC. An easy way to access the job log is to configure the ODBC DSN you are using through **ODBC Adminstrator** by selecting the **Print job log at disconnect** option on the **Diagnostic** tab. To find the job log, open a **PC5250** emulation session and run the WRKSPLF command. Specify the iSeries user profile that was used on the ODBC connection as the user parameter for the WRKSPLF command.
QAQQINI (Query options file)	You can set the library for Query options file, by configuring the ODBC DSN you are using through **ODBC Adminstrator** and selecting the **Diagnostic** tab. Enter the name of the library you want to use in the Query options file library box.

Collecting an ODBC Trace (SQL.LOG):

Steps for collecting ODBC API calls

Follow these steps to collect an SQL.LOG:

1. Start **ODBC Data Source Administrator**.
2. Select the **Tracing** tab
3. Select the **Start Tracing Now** button.
4. Select **Apply** or **OK**.
5. Recreate the error
6. Return to **ODBC Administrator**.
7. Select the **Tracing** tab.
8. Select the **Stop Tracing Now** button.
9. The trace can be viewed in the location that you initially specified in the **Log file Path** box.

Note: This procedure applies when you are using MDAC version 2.5. If you are using a different version of MDAC, then you may need to follow different steps.

Gather a detailed trace:

ODBC items that are useful in this trace include entry points into the driver, information about the prestart job, package name in use, and special error conditions.

Note: There are steps that need to be done before getting a detail trace for Microsoft Transaction Server (MTS). Complete the steps to gather a detail trace for a Microsoft Transaction Server (MTS) before completing the steps below.

1. From the Start menu choose **Programs** → **IBM iSeries Access for Windows** → **iSeries Access for Windows Properties**.
2. Click the **Diagnostic Tools** tab.
3. Click the **Start Diagnostic Tools** button.
4. Click **OK**. In the right of your desktop, you will see an icon that looks like a computer with a red dot on it.
5. Right-click on the icon and choose **Start All Diagnostics**
6. Re-create the problem.
7. Right-click the icon and select **Detail trace** → **Stop**.
8. Right-click the icon and select **Detail trace** → **Display**.
9. From the File menu select **Save As**.
10. Type a name and click the **Save** button.

Gather a detail trace for a Microsoft Transaction Server (MTS):

Identify steps for gathering this trace.

1. Make sure that you have Incoming Remote Command (IRC), an iSeries Access for Windows optional component, installed on the machine that has MTS and Microsoft Distributed Transaction Coordinator (MSDTC).
2. Make sure that IRC is running in the same account that MSDTC is running. Verify them in Start/Settings/Control Panel/Services.
3. At a command prompt, run **REXEC dragonfire CWBLOG START/DETAILTRACE**. Replace ″dragonfire″ with your PC name.
4. IRC will ask for a userID and password. Enter a userID with administrator’s authority.
5. Complete the steps to gather a detail trace.

iSeries Access ODBC error messages

When an error occurs, the iSeries Access ODBC driver returns the SQLSTATE (an ODBC error code) and an error message. The driver obtains this information both from errors that are detected by the driver and from errors that are returned by the DBMS.

For errors that occur in the data source, the iSeries Access ODBC Driver maps the returned native error to the appropriate SQLSTATE. When both the iSeries Access ODBC driver and the Microsoft Driver Manager detect an error, they generate the appropriate SQLSTATE. The iSeries Access ODBC driver returns an error message based on the message returned by the DBMS.

For errors that occur in the iSeries Access ODBC driver or the Microsoft Driver Manager, the iSeries Access ODBC driver returns an error message based on the text associated with the SQLSTATE.

Error message format

Error messages have the following format:

```
[vendor][ODBC-component][data-source]
error-message
```

The prefixes in brackets ([]) identify the source of the error. The following table shows the values of these prefixes returned by the iSeries Access ODBC driver.

When the error occurs in the data source, the [vendor] and [ODBC-component] prefixes identify the vendor and name of the ODBC component that received the error from the data source.

Error source	**Value**

Driver Manager	`[Microsoft]`
	`[ODBC driver Manager]`
	`[N/A]`

iSeries Access ODBC driver	`[IBM(R)]`
	`[iSeries Access ODBC driver]`
	`N/A`

NLS messages	`[IBM]`
	`[iSeries Access ODBC driver]`
	`Column #:`
	`NLS error message number`
	`NLS error message text`

Communication layer	`[IBM]`
	`[iSeries Access ODBC driver]`
	
	`Communications link failure.Comm RC=xxxx - (message text) Where xxxx is the error number in decimal, not hexadecimal, format. Message text describing the nature of your error appears with the error number.`
	Note: For more information about error message ids, see iSeries Access return codes or the iSeries Access for Windows online User's Guide.

DB2 UDB for iSeries	`[IBM]`
	`[iSeries Access ODBC driver]`
	`[DB2 UDB]`
	`Server error message`

Viewing DB2 UDB for iSeries error message text:

For errors that begin with:	Use this CL command
SQL	DSPMSGD RANGE(SQLxxxx) MSGF(QSQLMSG)

IWS or PWS	DSPMSGD RANGE(ZZZxxxx) MSGF(QIWS/QIWSMSG) where ZZZ is IWS or PWS

Refer to Common ODBC errors for help with other ODBC error messages.

You can search and view NLS or communication error messages in the Service, Error and Trace message help topic in the iSeries Access for Windows online User's Guide.

Related concepts

iSeries Access return codes

"Common ODBC errors" on page 21
Find and resolve ODBC errors.

Troubleshoot the iSeries server connection

Each ODBC connection communicates with one database server program that runs on the iSeries server. This program is referred to as the **host server program**.

The name of the Database Server program used with TCP/IP is **QZDASOINIT**. It is normally located in subsystem QUSRWRK, however it can be set up differently by the system administrator.

Under normal conditions, the program is evoked transparently, and the user is not required to take action except to verify that the proper subsystems and communication protocols are running. See the Host server administration for details on administration of host server jobs.

The most common indication of a connection failure is an error message from the ODBC driver mentioning a communications link failure.

If ODBC is unable to connect to the iSeries server, perform the following troubleshooting tasks:

Related concepts

"Host server administration" on page 27
Describes the host servers that are commonly used with iSeries Access for Windows, and describes how to effectively manage and use them.

Checking the server status:

Use CWBPING.

The iSeries Access for Windows product has a special command to verify status of host servers:

```
CWBPING systemname
```

where `systemname` is the name of the system.

The command should return something like the following:

```
To cancel the CWBPING request, press CTRL-C or CTRL=BREAK
I - Verifying connection to system MYSYSTEM...
I - Successfully connected to server application: Central Client
I - Successfully connected to server application: Network File
I - Successfully connected to server application: Network Print
I - Successfully connected to server application: Data Access
I - Successfully connected to server application: Data Queues
I - Successfully connected to server application: Remote Command
I - Successfully connected to server application: Security
I - Successfully connected to server application: DDM
I - Successfully connected to server application: Telnet
I - Successfully connected to server application: Management Central
I - Connection verified to system MYSYSTEM
```

Related concepts

"ODBC diagnostic and performance tools" on page 15
Use tools to help diagnose ODBC problems.

Verifying that subsystems are active:

TCP/IP-connected ODBC jobs (QZDASOINIT) will run in the QUSRWRK subsystem. Verify that this subsystem is running.

The QSERVER subsystem may need to be manually started. To do this, simply issue the following command:

```
STRSBS QSERVER
```

To have the subsystem start automatically at IPL, modify the IPL Start up procedure (the default is QSYS/QSTRUP) to include the STRSBS QSERVER command.

In addition to subsystem QSERVER, subsystem QSYSWRK, and QUSRWRK must be running.

Verifying that prestart jobs are running:

IBM ships the QSERVER/QUSRWRK subsystems to use prestart jobs to improve performance at job initialization and startup.

When prestart jobs are configured in the subsystem, the job MUST be active to connect. The prestart job used for a TCP/IP connection is:

- QZDASOINIT - Server program

To verify a prestart job is running use one of the following:

```
WRKACTJOB SBS(QUSRWRK)

WRKACTJOB SBS('user-defined-subsystem')
```

The appropriate prestart job should be active:

```
Job           User      Type     -----Status-----
QZDASOINIT   QUSER      PJ       ACTIVE           (socket connection)
```

Prestart jobs do not display in WRKACTJOB unless a connection is already active. You must use F14 - Include from the WRKACTJOB panel.

Additional TCP/IP considerations:

Use NETSTAT, STRTCP, and STRHOSTSVR to verify and start TCP/IP functions.

Verify that TCP/IP is started with the following command:

```
NETSTAT *CNN
```

Note: To verify that TCP/IP is started with iSeries Navigator, you must already have configured your server with TCP/IP , then do the following:

1. In iSeries Navigator, select your **server** → **Network**.
2. Right-click TCP/IP Configuration, and select Utilities.
3. Select Ping.
4. Specify a host name or TCP/IP address, and click Ping Now.

Use the command STRTCP to start the desired protocol if it is not running.

Verify the necessary daemons are running by browsing the information returned from the NETSTAT *CNN command:

```
Remote          Remote     Local
Address         Port       Port       Idle Time  State
*               *          as-cent >  000:09:31  Listen
*               *          as-signon  000:09:41  Listen
*               *          as-svrmap  002:57:45  Listen
*               *          as-data >  002:57:45  Listen
```

Use the command STRHOSTSVR SERVER(*ALL) to start them if necessary.

- Verify QZDASRVSD, the ODBC socket daemon, is running in the QSERVER subsystem.
  - as-database should be in the Listen State
  - WRKJOB QZDASRVSD should be used to check the job log of the daemon for any error messages.
- Verify that socket daemon QZSOSMAPD is running in QSYSWRK subsystem.
  - as-svrmap should be in the Listen State as shown by NETSTAT *CNN.
  - WRKJOB QZSOSMAPD should be used to check the job log of the daemon for any error messages.

The PC locates the socket used by the database server by connecting to the server mapper socket. It retrieves the socket used by as-database. It then connects to the proper socket which is being monitored by the database server daemon, QZDASRVSD. The server daemon will attach the client's connection to a QZDASOINIT prestart job in QUSRWRK. If this is the first connection made to the server from this PC, then two other servers are used: Central server for licensing and signon server for userid/password validation.

For more information about verifying that TCP/IP is started, see General TCP/IP problems.

Related concepts

General TCP/IP problems

Related tasks

Configure your server with TCP/IP

Common ODBC errors

Find and resolve ODBC errors.

The following topics provide general guidelines for finding and resolving common iSeries Access for Windows ODBC errors:

Related concepts

"iSeries Access ODBC error messages" on page 18
When an error occurs, the iSeries Access ODBC driver returns the SQLSTATE (an ODBC error code) and an error message. The driver obtains this information both from errors that are detected by the driver and from errors that are returned by the DBMS.

SQL errors:

List of common SQL errors that are encountered by applications

Note: For more information on SQL errors, see SQL messages and codes.

Related concepts

SQL messages and codes

SQL0104 - Token &1 was not valid. Valid tokens: &2:

Invalid SQL Syntax message

Probable cause:

- The application generated an SQL statement with incorrect syntax. For help with problem determination, use the ODBC trace tool, provided with the ODBC Administrator, to look at the SQL.LOG.
- See SQL0114 - Relational database &1 not the same as current &2 server if "*" is the token.
- The SQL statement is using a literal that exceeds the 32K size limitation. Consider using a parameter marker instead of a literal. This reduces the size of the statement while allowing you to pass the maximum field size woth of data.
- The application is using incorrect syntax for left outer join. Some applications default to a proprietary left outer join syntax of *= in the WHERE clause (PowerBuilder 3.0 & 4.0, Crystal Reports). Check with your application vendor. Most provide an ini setting or a configuration value to use ODBC left outer join syntax.
- Your ODBC Data Source Name (DSN) configuration uses the wrong decimal separator character. Some users have set the decimal separator parameter of the ODBC connection to a comma instead of a period.

Related concepts

"SQL0114 - Relational database &1 not the same as current &2 server"
Update the Relational Database Directory Entry.

SQL0113 - Name &1 not allowed.:

Update the Relational Database Directory

Probable cause:

It is likely that the system name is not in the Relational Database Directory. Run the Add Relational Database Directory Entry command:

```
ADDRDBDIRE RDB(SYSNAME) RMTLOCNAME(*LOCAL)
```

In the above example, SYSNAME is the name of your system's Default Local Location name (as specified in the DSPNETA command).

Another common cause for this error is a period (.) in a table or library name. Although the period is valid in i5/OS file naming conventions the name must be enclosed in double quotes to be used in a SQL statement. A short term circumvention may be to build a logical file over the desired physical file, using the SQL naming syntax. Another possible solution is to create an SQL Alias over the desired file and then access the file indirectly through the alias.

SQL0114 - Relational database &1 not the same as current &2 server:

Update the Relational Database Directory Entry.

Probable cause:

It is likely that the system name is not in the Remote Database Directory. Run the Add Relational Database Directory Entry command:

```
ADDRDBDIRE RDB(SYSNAME) RMTLOCNAME(*LOCAL)
```

In this above example, SYSNAME is the name of your system's Default Local Location name (as specified in the DSPNETA command).

Another common cause for this error is a period (.) in a table or library name. Although valid in naming conventions, in order to use it within an SQL statement, enclose the name within double quotes. A short term circumvention may be to build a logical file over the desired physical file, using the SQL naming syntax.

Related concepts

"SQL0104 - Token &1 was not valid. Valid tokens: &2" on page 21
Invalid SQL Syntax message

SQL0204 - MYSYSCONF not found:

Optional table on the server.

Probable cause:

Usually only job logs for jobs using the Microsoft Jet Engine (Microsoft ACCESS or Microsoft Visual Basic applications) contain this message. The MS Jet Engine always checks for an optional table on the server that is called MYSYSCONF. The applications ignore this warning. For further information, see the Microsoft Jet Database Engine Connectivity white paper or contact Microsoft.

SQL0208 - ORDER BY column not in result table:

Problem with ORDER BY clause

Probable cause:

The iSeries Access ODBC driver reports ″Y″ to the property SQL_ORDER_BY_COLUMNS_IN_SELECT (ODBC 2.0). A character string of ″Y″ implies that the columns in the ORDER BY clause must be in the select list. Some common desktop reporting applications either ignore or do not check this value and attempt to use an order by field which is not in the select list.

SQL0900 - Application process not in a connected state:

Update the Relational Database Directory Entry.

Probable cause:

It is likely that the system name is not in the Remote Database Directory. Run the Add Relational Database Directory Entry command:

```
ADDRDBDIRE RDB(SYSNAME) RMTLOCNAME(*LOCAL)
```

In the above example, SYSNAME represents the name of your system's Default Local Location name (as specified in the DSPNETA command).

Another common cause for this error is a period (.) in a table or library name. Although valid in naming conventions, in order to use it within an SQL statement, enclose the name within double quotes. A short term circumvention may be to build a logical file over the desired physical file, using the SQL naming syntax.

Your ODBC Data Source Name (DSN) configuration uses the wrong naming convention. Use the ODBC Administrator to change your DSN to use the proper (*SQL or *SYS) naming convention. Always use *SQL unless your application design specifically expects *SYS.

SQL0901 - SQL System Error:

Server machine (function) check error

Probable cause:

Another, previously reported error has prevented the processing of a SQL statement. The previous error is logged only in the i5/OS job log and is not returned to the ODBC application. You must locate and retrieve the job log to identify and resolve the problem.

To find the job log, open a PC5250 emulation session and issue a WRKSPLF where user is the iSeries user profile used on the ODBC connection. However, in some cases the joblog is found using WRKSPLF QUSER. For example, it is necessary to use WRKSPLF QUSER to find the associated joblog when the prestart jobs fail to start.

SQL5001 - Column qualifier or table &2 undefined.:

Change your naming convention in your ODBC DSN.

Probable cause:

Your ODBC Data Source Name (DSN) configuration uses the wrong naming convention. Use the ODBC Administrator to change your DSN to use the proper (*SQL or *SYS) naming convention. Always use *SQL unless your application design specifically expects *SYS.

SQL5016 - Object name &1 not valid for naming convention:

Change your naming convention in your ODBC DSN.

Probable cause:

Your ODBC Data Source Name (DSN) configuration uses the wrong naming convention. Use the ODBC Administrator to change your DSN to use the proper (*SQL or *SYS) naming convention. Always use *SQL unless your application design specifically expects *SYS.

SQL7008 - &1 in &2 not valid for operation. The reason code is 3:

Error related to files not journaled

Probable cause:

The database performs commitment control by journaling. Any ODBC application that takes advantage of commitment control will require journaling the files that are used.

Stored procedure errors:

Common stored procedure errors returned to applications

SQL0444 - External program &A in &B not found (DB2 UDB for iSeries SQL):

The SQL0444 is generated on an execute or execute direct when the database server is able to locate the procedure declaration but is unable to locate the program object.

The external program must be in the location specified in the system catalog tables. Note that this location is defined by the naming convention and default collection in affect when the procedure is defined (using CREATE PROCEDURE) and not when the procedure is called. To check the location defined for the external program name of a stored procedure run a query over QSYS2.SYSPROCS and note the value for the "EXTERNAL_NAME" name field.

No data returned on OUTPUT and INPUT_OUTPUT parameters:

SQLBindParameter problem when no data returned

This problem could be caused by any of the following:

- The ODBC **SQLBindParameter** API incorrectly specified **fParamType** as SQL_PARAM_INPUT.
- DECLARE PROCEDURE was used instead of CREATE PROCEDURE, and extended dynamic support is disabled.
- The programmer incorrectly declared a parameter as IN on the CREATE or DECLARE PROCEDURE.
- The stored procedure program incorrectly returned the parameter.

SQL0501 - Cursor CRSR000x not open:

To return data when using embedded SQL in ILE programs, you must specify the compile option ACTGRP(*CALLER) and not the default of *NEW.

Verify that the program executes a return instead of an exit.

When the stored procedure program executes an exit instead of a return, you must set the **Close SQL Cursor** option to *ENDACTGRP. If the Close SQL Cursor option is set to *ENDMOD, the cursor will be closed before data is retrieved.

Also, verify that the CREATE PROCEDURE specifies the correct number of result sets. This is especially important when using array result sets.

ODBC incorrect output and unpredictable errors:

Ensure that the iSeries Access ODBC driver and the database server program are at matching code levels.

Check for PTF corequisite requirements on any PTF that you order or in the readme.txt file of the Service Pack. If problems continue, verify that you have disabled the prefetch option in the ODBC Data Source. The prefetch option should not be used if the application uses either the SQLExtendedFetch or SQLFetchScroll ODBC API, or if you are not sure.

Note that *result set cursors* from stored procedures are forward only, read only.

Note: Binary or hexadecimal data instead of ASCII characters

> The default value of the Translation parameter is set to not convert binary data (CCSID 65535) to text. A CCSID is attached to files, tables, and even fields (columns) to identify the conversion table that is used to convert the data. A CCSID of 65535 often identifies raw data (binary or hexadecimal), such as bitmapped graphics, that is language independent. Not selecting *Convert binary data (CCSID 65535) to text* ensures that the raw data is not damaged.
>
> Setting the Translation parameter to *Convert binary data (CCSID 65535) to text*, changes the CCSID that is attached to the data to the CCSID that is attached to the job. **This parameter setting can cause damage to the data, if the data is truly binary.**

Gather information for IBM Support

The IBM Support staff can offer you better service, if you have certain information available when you open a problem record to IBM Support.

To gather this information, complete the following tasks:

Run **cwbsvget.exe** to gather information.	The **cwbsvget.exe** tool, a part of iSeries Access for Windows V5R3 and later, can help collect all traces run plus other information that may be helpful in diagnosing a problem. **cwbsvget** produces a zip file to send to IBM Service for analysis. Note that **cwbsvget** does NOT turn traces on and off -- it simply gathers traces and other data into one file for convenience and completeness. If you use the **cwbsvget.exe** tool you will not need to complete the steps below for gathering the version of the ODBC driver and for locating the trace files. Make sure to run **cwbsvget.exe** after the traces are stopped so that the trace files get packaged into the zip file that **cwbsvget** generates. To use **cwbsvget.exe** complete the following steps: 1. Open a MS DOS Command prompt. 2. Navigate to the Client Access folder typically located in the \Program Files\IBM\Client Access directory and run the following command: `cd \Program Files\IBM\Client Access` 3. Run the command: **cwbsvget.exe** **Note:** **cwbsvget.exe** generates a .zip file for you. The output on the DOS Command window indicates where that .zip file was created.
Record the i5/OS version and cumulative PTF level.	1. Issue the display PTF command on an terminal emulation command line: `DSPPTF` 2. Record the i5/OS release information that has the format VxRxMx. 3. Verify that the IPL source is ##MACH#B. 4. Press **F5** to display the PTF details. 5. Record the first PTF ID in the list. It will have the format Tzxxyyy where xx is the year, yyy the Julian date and z is either L or C.
Record the version of the ODBC driver.	1. From the Task bar select **Start** → **Programs** → **IBM iSeries Access for Windows** → **ODBC Administration**. **Note:** On a 64-bit machine using a 64-bit driver, select **ODBC Administration (64-bit)**. 2. Select the **Drivers** tab. 3. Record the version of the iSeries Access ODBC Driver.
Record the version of the ODBC driver manager.	1. From the Task bar select **Start** → **Programs** → **IBM iSeries Access for Windows** → **ODBC Administration**. **Note:** On a 64-bit machine using a 64-bit driver, select **ODBC Administration (64-bit)**. 2. Select the **About** tab. 3. Record the version of the Driver Manager.

Gather traces	The traces you will most likely be asked to gather for support are: an ODBC trace (SQL.LOG), CWBCOTRC or Communication Trace, and a Detail Trace. See ODBC diagnostic and performance tools, for more information about traces.
Record additional information	Such as the PC application, the error description, and what ODBC driver (32-bit or 64-bit) you are using.

Related concepts

"ODBC diagnostic and performance tools" on page 15
Use tools to help diagnose ODBC problems.

Host server administration

Describes the host servers that are commonly used with iSeries Access for Windows, and describes how to effectively manage and use them.

This topic provides brief descriptions of server functions that run on an iSeries server and technical information specific to host servers that are used by the iSeries Access for Windows product. These are not all of the servers used by iSeries Access for Windows, and this topic does not address all of the servers on the host (iSeries) system.

i5/OS 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. iSeries 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 becomes complex. Having all of your servers on one integrated system greatly reduces the overall cost and complexity of managing your network.

These servers are used by iSeries Access for Windows, but are designed so that other client products can also use them. This topic focuses on how these servers are used by iSeries Access for Windows.

Adding or removing the Host Server option

The servers discussed here are all optimized servers, and are included with the base option of i5/OS. To use the iSeries Navigator function of iSeries Access for Windows, install the Host Server option.

If you are not using any iSeries Access for Windows products or iSeries 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.

Related concepts

"ODBC program security strategies" on page 13
Consider the following ODBC program security strategies.

"Troubleshoot the iSeries server connection" on page 19
Each ODBC connection communicates with one database server program that runs on the iSeries server. This program is referred to as the **host server program**.

Identify i5/OS host servers and associated programs.

Describes many of the host servers that are common in the iSeries Access for Windows client and the related objects. You can view the servers by type or by their function in iSeries Access for Windows.

This information covers only the servers used by iSeries Access for Windows. This does not include all of the servers on the host (iSeries) system. iSeries Access for Windows host servers include:

Host servers by iSeries Access for Windows function

Host servers listed by their associated function in iSeries Access for Windows.

The following table shows a subset of the servers that are used with some of the functions in iSeries Access for Windows.

Client function	i5/OS server used
.NET Data Provider	• Database Server • Signon server • Central server • QXDAEDRSQL server
IBM Toolbox for Java™	• 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	• Signon server • Central server • Database server
ODBC driver	• Signon server • Database server
Access integrated file system from iSeries Navigator	File server
Data queue APIs	Data queue server
OLE DB provider	• Data queue server • Database server • Remote command and distributed program call server • Signon server
Extended Dynamic Remote SQL server (QXDAEDRSQL)	• 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
Retrieve conversion map Done only on initial connection if the client does not contain the required conversion maps	Central server

Client function	i5/OS server used
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 (TCP/IP)	Signon server
Network Print	Network print server

For more information, refer to iSeries Access for Windows Servers and Ports Required, APAR II12227.

Related information

APAR II12227

File server

Learn about the file server, including file server programs and how it works with the integrated file system.

The integrated file system is a part of the base iSeries server operating system that supports stream input/output and storage management, similar to personal computer and UNIX operating systems. The integrated file system also integrates all information that is stored on the iSeries server. iSeries servers can support several different file systems with similar interfaces. A file system allows users and applications to access specific segments of storage that are organized as file, directory, library, and object logical units.

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

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 iSeries 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 iSeries server.

For a list of iSeries 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

Work with file systems

Integrated file system

File server programs:

See a list of 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
QPWFSERVSO	QSYS	*PGM	Server program
QPWFSERVS2	QSYS	*PGM	Server program
QPWFSERVSD	QSYS	*PGM	Daemon program
QPWFSERV	QSYS	*JOBD	Job description used for server jobs
QPWFSERVER	QSYS	*CLS	Class used for all file server and database server jobs
QPWFSERVSS	QSYS	*PGM	SSL server program

Database server

For Data Transfer, ODBC, iSeries Navigator database, and iSeries Access for Windows providers (OLE DB and the .NET Data provider).

The database server allows clients access to the functions included with **DB2 UDB for iSeries** . This server provides:

- 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 concepts

Distributed database programming

Database server programs:

See a list of 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:

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, see:

SQL package names:

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 server.

Package names if the relational database is not an iSeries server

The package is created in a collection called QSQL400 on the application server if the relational database (RDB) is not an iSeries server. When the application server is not an iSeries server, the package name is QZD **abcde**, in which **abcde** corresponds to specific parser options being used.

If the RDB is an iSeries server, the package is usually created in the QGPL library which most database access clients can customize.

The following table shows the options for the package name.

Package name field options

Field	Field description	Options
a	Date format	• ISO, JIS • USA • EUR • JUL
b	Time format	• JIS • USA • EUR, ISO
c	Commitment control/ decimal delimiter	• *CS/period • *CS/comma • *CHG/period • *CHG/comma • *RR/period • *RR/comma
d	String delimiter	• apostrophe • quote
e	Maximum number of statements allowed for package	• 0 - 64 • 1 - 256 • 2 - 512 • 3 - 1024

Package names if the relational database is an iSeries server

When the application server is an iSeries server, the package name is QZDA **abcdef**, in which **abcdef** corresponds to specific parser options being used.

Package name field options

Field	Field description	Options
a	Date format	• ISO, JIS • USA • EUR • JUL • MDY • DMY • YMD
b	Time format and naming convention	• 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	• *CS/period • *CS/comma • *ALL/period • *ALL/comma • *CHG/period • *CHG/comma • *NONE/period • *NONE/comma
d	String delimiter	• apostrophe • quote
e	Number of sections in package	• 0 - 64 • 1 - 256 • 2 - 512 • 3 - 1024
f	Date and Time separation	• 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:

The packages used for DRDA functions are created automatically on your system as needed. You might want to periodically cleanup these packages. To delete the packages, use the Delete SQL Package (DLTSQLPKG) 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.

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 iSeries naming convention, although the format of STMTxxxx is suggested Cursor name must adhere to iSeries naming conventions	Statement name must adhere to iSeries naming convention, although the format of STMTxxxx is suggested Cursor name must adhere to iSeries 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 an iSeries 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:

Distributed Relational Database Architecture (DRDA) is an architecture that allows access to other databases that support DRDA. For more information about DRDA, see Distributed database programming.

When using the database server as a gateway to other RDBs using DRDA, some limitations in functions must be followed.

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

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	• 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 connected to an iSeries server
Commit level *NONE	Not supported
Commit level *CHANGE	Only supported if the target RDB is an iSeries. All other RDBs require a *CS or *ALL commit level.

Related concepts

Distributed database programming

Data queue server

Provides access to iSeries Server data queues.

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

iSeries Access for Windows provides APIs that can allow PC applications to work with iSeries data queues with the same ease that iSeries applications can. This extends iSeries 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 functions.

The network print server allows enhanced client control over print resources on the iSeries server. 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 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 iSeries Access for Windows 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 iSeries Navigator to view the iSeries 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
QZSCSRVS	QSYS	Server program
QZSCSRVSD	QSYS	Daemon program

Remote command and distributed program call server

Allows PC applications to issue commands and call programs on i5/OS and return the results to the client.

The remote command and distributed program call server support allows users and applications to issue iSeries 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 iSeries users with limited capabilities (LMTCPB =*YES, in their user profile).

The distributed program call support allows applications to call iSeries programs and pass parameters (input and output). After the program runs on the iSeries server, the output parameter values return to the client application. This process allows applications to access iSeries 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
QZRCSRVSD	QSYS	Daemon program

Signon server

Provides password management functions for 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 iSeries 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 a 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.

Server port mapper

Program name	Library	Description
QZSOSMAPD	QSYS	Server port mapper program

Extended Dynamic Remote SQL server (QXDAEDRSQL)

Supports remote SQL access and other database functions.

The QXDAEDRSQL server allows clients access to the functions included with DB2 UDB for iSeries. This server provides:

- 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 QXDAIASP *SRVPGM objects are used by the QXDAEDRSQL server.

DRDA/DDM server

Allows access to functions included with DB2 UDB for iSeries. This server supports record level access when using the OLE DB provider and the Toolbox record level access classes.

The DRDA/DDM server allows clients access to the functions included with DB2 UDB for iSeries, 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	Listerner program

Related concepts
Distributed database programming
Distributed data management

Use i5/OS host servers

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

The servers shipped with the base operating system do not typically require any changes to your existing system configuration in order to work correctly. They are set up and configured when you install the i5/OS 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 concepts

Work management

Establish client/server communications

Learn the process for starting and ending communication between 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 i5/OS 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

iSeries Access for Windows uses TCP/IP to communicate with the iSeries system servers. The optimized servers use i5/OS sockets support to communicate with clients. The i5/OS sockets support is compatible with Berkeley Software Distributions 4.3 sockets over TCP/IP. Sockets support is provided with the 5722-TC1 product that is installed on the iSeries server.

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 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 iSeries 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 each server's port number for the optimized servers and server mapper that use sockets over TCP communication support and those that use Secure Sockets Layer (SSL).

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

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	DDM	446
as-admin-http	HTTP administration	2001
as-mtgctrlj	Management central	5544
as-mtgctrl	Management central	5555
telnet	Telnet server	23
as-edrsql	QXDAEDRSQL server	4402

The following table shows port numbers for host servers and daemons that use Secure Sockets Layer (SSL):

Service name	**Description**	**Port Number**
as-central-s	Secure central server	9470
as-database-s	Secure database server	9471
as-dtaq-s	Secure data queue server	9472
as-file-s	Secure file server	9473
as-netprt-s	Secure network print server	9474

Service name	Description	Port Number
as-rmtcmd-s	Secure remote command/ Program call server	9475
as-signon-s	Secure signon server	9476
ddm-ssl	DDM	448
as-admin-https	HTTP administration	2010
as-mgtctrlj	Management central	5544
as-mgtctrl-ss	Management central	5566
as-mgtctrl-cs	Management central	5577
Telnet-ssl	Telnet server	992

Note: For more information, see CWBCO1003, in the iSeries Access for Windows online User's Guide (on the contents tab select, **Messages** → **iSeries Access for Windows Messages** → **CWBCO1003)**.

Display and Modify Service Table Entries

You can use the WRKSRVTBLE command to display the service names and their associated port numbers.

```
+--------------------------------------------------------------------------------+
|                   Work with Service Table Entries                              |
|                                                           System:   AS400597   |
| Type options, press Enter.                                                     |
|   1=Add   4=Remove   5=Display                                                 |
|                                                                                |
| Opt  Service                                 Port  Protocol                    |
|  _   ________________________________  ____________ _________________________  |
|  _   as-central                              8470  tcp                         |
|  _   as-database                             8471  tcp                         |
|  _   as-dtaq                                 8472  tcp                         |
|  _   as-file                                 8473  tcp                         |
|  _   as-netprt                               8474  tcp                         |
|  _   as-rmtcmd                               8475  tcp                         |
|  _   as-signon                               8476  tcp                         |
|  _   as-svrmap                                449  tcp                         |
|         .                                                                      |
|         .                                                                      |
|         .                                                                      |
|                                                                                |
+--------------------------------------------------------------------------------+
```

By selecting option 5 (display) for any entry, you also see the alias names. Use the ADDSRVTBLE and RMVSRVTBLE commands to change the service table for your installation.

Start host servers:

To start Host Servers, use the STRHOSTSVR CL command.

Note: You can use iSeries Navigator to configure your system so that servers start automatically when you start Transmission Control Protocol (TCP) with the STRTCP command. Newly shipped systems do this by default.

The STRHOSTSVR command starts the host server daemons and the server mapper daemon. It also attempts to start the prestart job associated with the server.

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 QNPSERVD, and the associated server prestart jobs are QNPSERVS 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 QZRCSRVSD, 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 concepts

CL command finder

Server daemons:

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 ENDHOSTSVR command.

Example: STRHOSTSVR:

Find examples of using the STRHOSTSVR command.

Example 1: Starting all host server daemons

```
STRHOSTSVR(*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 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.

Subsystems on the iSeries server

Describes which system-supplied subsystems are used for each of the server functions. These topics also detail how the subsystem descriptions relate to the server jobs. Learn about i5/OS subsystems and how to autostart and prestart jobs.

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 server jobs on the iSeries server" on page 59
Learn ways to identify and display server jobs.

"Use the character-based interface to display server job" on page 59
Learn how to display server jobs using the character-based interface.

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 of 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.

Beginning in V5R1, 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. When

TCP/IP is started, it ensures QUSRWRK and QSERVER are started before starting the host servers. If slip installing V5R1 (or later) on a system that was at a release prior to V5R1, and if the startup program used by the system had been changed to start TCP/IP, then the system will automatically start TCP/IP, and the startup program's attempt will fail. 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:

Learn about the autostart jobs associated with the use of the 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 of 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 in order to start using TCP/IP:

1. ENDHOSTSVR *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(*JOBD) 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 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:

- Prestart jobs can be reused, but there is no automatic cleanup for the prestart job once it has been used and subsequently 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. This means that resources that are used by one user of the prestart job must be cleaned up before ending use of the prestart job. Otherwise, these resources will maintain the same status for the next user that uses the prestart job. For example, a file that is opened but never closed by one user of a prestart job remains open and available to the following user of the same prestart job.

- By default, some of the server jobs run in QUSRWRK or QSERVER. Using iSeries Navigator, you can configure some or all of these servers to run in a subsystem of your choice.
  1. Double-click **iSeries Navigator** → **Network** → **Servers** → **iSeries 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/QNPSERVS
Remote command and program call	QSYS/QZRCSRVS
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/QZBSJOBD
Remote command and program call	QSYS/QZBSJOBD
Central	QSYS/QZBSJOBD
Database	QGPL/QDFTSVR
Secure Database	QGPL/QDFTSVR
File	QGPL/QDFTSVR
Secure File	QGPL/QDFTSVR
Data Queue	QSYS/QZBSJOBD
Signon	QSYS/QZBSJOBD

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
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

Host server	Value
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
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

Host server	Value
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/QCASERVR
Remote command and program call	QGPL/QCASERVR
Central	QGPL/QCASERVR
Database	QSYS/QPWFSERVER
Secure Database	QSYS/QPWFSERVER
File	QSYS/QPWFSERVER
Secure File	QSYS/QPWFSERVER
Data Queue	QGPL/QCASERVR
Signon	QGPL/QCASERVR

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:

- When the subsystem is started, one prestart job for each server is started.
- When the first client connection request processes for a specific server, the initial job is used and the threshold is exceeded.
- Additional jobs are started for that server based on the number that is defined in the prestart job entry.
- The number of available jobs is always at least one.
- The subsystem periodically checks the number of prestart jobs that are ready to process requests, and ends excess jobs. The subsystem always leaves at least the number of prestart jobs specified in the initial jobs parameter.

Monitor prestart jobs

Use the Display Active Prestart Jobs (DSPACTPJ) command to monitor the prestart jobs. For example, to monitor prestart jobs for the signon server, you must know the subsystem your prestart jobs are in (QUSRWRK or a user-defined subsystem) and the program (for example, QZSOSIGN).

The DSPACTPJ command provides the following information:

```
+--------------------------------------------------------------------------------+
|                   Display Active Prestart Jobs                AS400597          |
|                                                     01/12/95  16:39:25          |
|Subsystem  . . . . . :   QUSRWRK       Reset date . . . . . :   01/11/95          |
|Program  . . . . . . :   QZSOSIGN      Reset time . . . . . :   16:54:50          |
|  Library  . . . . . :     QSYS      Elapsed time . . . . :   0023:12:21          |
|                                                                                |
| Prestart jobs:                                                                 |
|   Current number . . . . . . . . . . . . . . . . . :   10                       |
|   Average number . . . . . . . . . . . . . . . . . :   8.5                      |
|   Peak number  . . . . . . . . . . . . . . . . . . :   25                       |
|                                                                                |
| Prestart jobs in use:                                                          |
|   Current number . . . . . . . . . . . . . . . . . :   5                        |
|   Average number . . . . . . . . . . . . . . . . . :   4.3                      |
|   Peak number  . . . . . . . . . . . . . . . . . . :   25                       |
|                                                                                |
|                                                                                |
|                                                            More...             |
|                                                                                |
|                                                                                |
+--------------------------------------------------------------------------------+
```

```
+--------------------------------------------------------------------------------+
|                                                     01/12/95  16:39:25          |
|Subsystem  . . . . . :   QUSRWRK       Reset date . . . . . :   01/11/95          |
|Program  . . . . . . :   QZSOSIGN      Reset time . . . . . :   16:54:50          |
|  Library  . . . . . :     QSYS      Elapsed time . . . . :   0023:12:21          |
|                                                                                |
|                                                                                |
|                                                                                |
| Program start requests:                                                        |
|   Current number waiting . . . . . . . . . . . . . :   0                        |
|   Average number waiting . . . . . . . . . . . . . :   .2                       |
|   Peak number waiting  . . . . . . . . . . . . . . :   4                        |
|   Average wait time  . . . . . . . . . . . . . . . :   00:00:20.0               |
|   Number accepted  . . . . . . . . . . . . . . . . :   0                        |
|   Number rejected  . . . . . . . . . . . . . . . . :   0                        |
|                                                                                |
|                                                                                |
|                                                            Bottom              |
|Press Enter to continue.                                                        |
|                                                                                |
|F3=Exit   F5=Refresh   F12=Cancel   F13=Reset statistics                        |
|                                                                                |
+--------------------------------------------------------------------------------+
```

Manage prestart jobs

Pressing the **F5** key while on the Display Active Prestart Jobs display can refresh the information presented for an active prestart job. The information about program start requests can indicate whether 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 QPWFSERVER 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

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

Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZRCSRVSD
Job name	QZRCSRVSD
Class	QGPL/QCASERVR
Sequence number	2539

Central server daemon

Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX
User	QUSER
Route data	QZSCSRVSD
Job name	QZSCSRVSD
Class	QGPL/QCASERVR
Sequence number	2536

Database server daemon

Subsystem	QSYS/QSERVER
Job queue	QPWFSERVER
User	QUSER
Route data	QZDASRVSD
Job name	QZDASRVSD
Class	QSYS/QPWFSERVER
Sequence number	600

File server daemon

Subsystem	QSYS/QSERVER
Job queue	QPWFSERVER
User	QUSER
Route data	QPWFSERVSD
Job name	QPWFSERVSD
Class	QSYS/QPWFSERVER
Sequence number	200

Data queue server daemon

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

Signon server daemon

Subsystem	QSYS/QSYSWRK
Job queue	QSYSNOMAX

User	QUSER
Route data	QZSOSGND
Job name	QZSOSGND
Class	QGPL/QCASERVR
Sequence number	2540

Server Mapper daemon

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

System values on the iSeries server

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.

The iSeries server has 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 the iSeries server will support, the type of encryption used for passwords, and whether

iSeries NetServer passwords for the Windows clients will be removed from the system. Changes that are made to this system value take effect on the next IPL.

Attention: If you set this value to support long passwords, you must upgrade all client PCs for long password support (Express V5R1) before setting this value. Otherwise, all pre-V5R1 clients will be unable to log onto the iSeries server.

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 work station function. When QRMTSIGN is set to *REJECT, the user can still sign-on to the system by using TELNET. These sessions will go through normal processing. 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 valid for passthrough.

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 server jobs on the iSeries server

Learn 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 iSeries Navigator interface to identify your server jobs if you prefer using a graphical user interface (GUI). You may 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

"Subsystems on the iSeries server" on page 44
Describes which system-supplied subsystems are used for each of the server functions. These topics also detail how the subsystem descriptions relate to the server jobs. Learn about i5/OS subsystems and how to autostart and prestart jobs.

Use iSeries Navigator to identify server jobs:

Learn how to display server jobs using iSeries Navigator.

Follow these steps to use the iSeries Navigator interface to identify your server jobs.

1. Double-click the **iSeries Navigator** icon.
2. Open **Network** by clicking the **plus sign (+)**.
3. Open **Servers** by clicking the **plus sign (+)**.
4. Select the type of servers for which you want to see jobs (For example, TCP/IP or iSeries Access for Windows).
5. 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.

Use the character-based interface to display server job:

Learn how to display server jobs using the character-based interface.

Choose from the following for information on how to identify server jobs using the traditional character-based interface:

Related concepts

"Subsystems on the iSeries server" on page 44
Describes which system-supplied subsystems are used for each of the server functions. These topics also detail how the subsystem descriptions relate to the server jobs. Learn about i5/OS subsystems and how to autostart and prestart jobs.

iSeries job names:

Learn how jobs are named on the iSeries

The job name that is used on the iSeries 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 nonprestarted jobs, 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 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                AS400597          |
|                                                    01/12/95  10:25:40          |
|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               |
|           .                                                                    |
|___    QSYSWRK       QSYS       SBS      .0                   DEQW               |
|___ (1) QZSOSMAPD    QUSER      BCH      .0                   SELW               |
|           .                                                                    |
|___ (2) QZSOSGND     QUSER      BCH      .0                   SELW               |
|___     QZSCSRVSD    QUSER      BCH      .0                   SELW               |
|___     QZRCSRVSD    QUSER       BCH      .0                    SELW            |
|___     QZHQSRVD     QUSER       BCH      .0                    SELW            |
|___     QNPSERVD     QUSER       BCH      .0                    SELW            |
|           .                                                                    |
|           .                                                                    |
|___   QUSRWRK        QSYS        SBS      .0                    DEQW            |
|___ (3) QZSOSIGN     QUSER       PJ       .0                    PSRW            |
|___     QZSCSRVS     QUSER       PJ       .0                    PSRW            |
|___     QZRCSRVS     QUSER       PJ       .0                    PSRW            |
|___     QZHQSSRV     QUSER       PJ       .0                    PSRW            |
|___     QNPSERVS     QUSER       PJ       .0                    PSRW            |
|___     QZDASOINIT   QUSER       PJ       .0                    PSRW            |
|           .                                                             More... |
+--------------------------------------------------------------------------------+
```

```
+--------------------------------------------------------------------------------+
|                         Work with Active Jobs              AS400597             |
|                                                  01/12/95 10:25:40             |
|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=Displaymessage        |
|  8=Work with spooled files   13=Disconnect ...                                 |
|                                                                                |
|Opt  Subsystem/Job  User     Type  CPU %  Function         Status               |
|           .                                                                    |
|___   QSERVER        QSYS        SBS      .0                   DEQW             |
|        QSERVER      QPGMR       ASJ      .1                   EVTW             |
|           .                                                                    |
|___ (2) QPWFSERVSD   QUSER       BCH      .0                   SELW             |
|        QZDASRVSD    QUSER       BCH      .0                   SELW             |
|           .                                                                    |
|           .                                                                    |
|___ (3) QPWFSERVSO   QUSER       PJ       .0                   PSRW             |
|___     QPWFSERVSO   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:

Learn how to find 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 iSeries Navigator or the WRKOBJLCK command.

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

1. Open **iSeries Navigator** (double-click on the icon).
2. Click on **Users and Groups**, then **All 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 EZ-Setup and iSeries Navigator with host servers

Learn how to tell if the required communication path is active, and how to start it if necessary.

EZ-Setup and iSeries Navigator may connect to the signon, central, and remote command and distributed program call servers without a communication protocol running on the iSeries server. That is, EZ-Setup may connect before STRTCP has been run. The path used permits EZ-Setup to perform some initial iSeries setup before configuring or starting any communication protocols. This topic describes how to determine if the communication path used by EZ-Setup and Operations Console is active and how to restart it if necessary.

For information on configuring the connection that is used by EZ-Setup consult the EZ-Setup online help.

The communication path used by EZ-Setup requires three jobs, QNEOSOEM, to be running in the QSYSWRK subsystem. The QSYSWRK subsystem has an autostart job for this communication path. The autostart job, QNEOSOEM, submits two other jobs with the name of QNEOSOEM in the QSYSWRK subsystem. If one of the jobs is not active, start it by issuing the following command:

```
QSYS/SBMJOB CMD(QSYS/CALL PGM(QSYS/QNEOSOEM)) JOB(QNEOSOEM)
JOBD(QSYS/QNEOJOBD) JOBQ(QSYS/QSYSNOMAX) PRTDEV(*JOBD) OUTQ(*JOBD)
USER(*JOBD) PRTTXT(*JOBD) SYSLIBL(*SYSVAL) INLLIBL(*JOBD)
LOGCLPGM(*YES) MSGQ(*NONE) SRTSEQ(*SYSVAL) LANGID(*SYSVAL)
CNTRYID(*SYSVAL) CCSID(*SYSVAL)
```

The command will start all three QNEOSOEM jobs if necessary.

Use server exit programs

Shows how to write and register exit programs. You can also find exit program parameters and programming examples in this topic.

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 148.

Register exit programs

Identify an exit program to call.

Work with the registration facility

In order 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 i5/OS 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                             |
|                                                                                |
|                          Exit                                                  |
|      Exit                Point                                                 |
| Opt  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                                    |
|                                                                                |
|              Exit                                                              |
|            Program     Exit                                                    |
| Opt         Number     Program        Library                                  |
| 1_                     __________     __________                               |
|                                                                                |
|   (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 affect.

```
+--------------------------------------------------------------------------------+
|                          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 provides the exit point and format names for each of the specific i5/OS servers.

QIBM_QPWFS_FILE_SERV (File Server)

Format Name	PWFS0100
Application Name	*FILESRV

QIBM_QZDA_INIT (Database server initiation)

Format Name	ZDAI0100
Application Name	*SQL

QIBM_QZDA_NDB1 (Database server-native database requests)

Format Names	ZDAQ0100 ZDAQ0200
Application Name	*NDB

QIBM_QZDA_ROI1 (Database server retrieve object information requests)

Format Names	ZDAR0100 ZDAR0200
Application Name	*RTVOBJINF

QIBM_QZDA_SQL1 (Database server SQL requests)

Format Names	ZDAQ0100
Application Name	*SQLSRV

QIBM_QZDA_SQL2 (Database server SQL requests)

Format Names	ZDAQ0200
Application Name	*SQLSRV

QIBM_QZHQ_DATA_QUEUE (Data queue server)

Format Name	ZHQ00100
Application Name	*DATAQSRV

QIBM_QNPS_ENTRY (Network print server)

Format Name	ENTR0100
Application Name	QNPSERVR

QIBM_QNPS_SPLF (Network print server)

Format Name	SPLF0100
Application Name	QNPSERVR

QIBM_QZSC_LM (Central server license management requests)

Format Name	ZSCL0100
Application Name	*CNTRLSRV

QIBM_QZSC_NLS (Central server NLS requests)

Format Name	ZSCN0100
Application Name	*CNTRLSRV

QIBM_QZSC_SM (License server)

Format Name	ZSCS0100
Application Name	*CNTRLSRV

QIBM_QZRC_RMT (Remote command and distributed program call server)

Format Name	CZRC0100
Application Name	*RMTSRV

QIBM_QZSO_SIGNONSRV (Signon server)

Format Name	ZSOY0100
Application Name	*SIGNON

Write exit programs

This topic identifies considerations when specifying an exit program.

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.

Exit program parameters 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 exit points for the servers.

Exit program parameters

Identify exit points for the servers.

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 an exit program.

File server:

Identify exit point for file server.

The file server has one exit point defined:

QIBM_QPWFS_FILE_SERV Format PWFS0100

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

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 will fail.

Exit point QIBM_QPWFS_FILE_SERV format PWFS0100

Offset				
Dec	**Hex**	**Type**	**Field**	**Description**
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: • **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: • 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 16MB. If the requested function has a value of **X'0007'** (Allocate conversation request), the file name length is 0.

Offset				
Dec	**Hex**	**Type**	**Field**	**Description**
40	28	CHAR(*)	File name	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. If a requested function has a value of one of the following, the file name is provided and the file name length is set: • **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
Notes: • This format is defined by member EPWFSEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC. • The APIs available to convert to and from CCSID 1200 are iconv() and CDRCVRT.				

Database server:

Identify exit points for database server.

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				
Dec	**Hex**	**Type**	**Field**	**Description**
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.

Offset		Type	Field	Description
Dec	Hex			
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.
Note: This format is defined by member EZDAEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

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.
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: • **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

Offset				
Dec	**Hex**	**Type**	**Field**	**Description**
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
Note: This format is defined by member EZDAEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Exit point QIBM_QZDA_NDB1 format ZDAD0200

Offset				
Dec	**Hex**	**Type**	**Field**	**Description**
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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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.
28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: • **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
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	• **0** - Connected to local RDB • **1** - Connected to remote RDB

Offset				
Dec	Hex	Type	Field	Description
94	5E	CHAR(1)	Commitment control level	• **’A’** - Commit *ALL • **’C’** - Commit *CHANGE • **’N’** - Commit *NONE • **’S’** - Commit *CS (cursor stability) • **’L’** - Commit *RR (repeatable read)
95	5F	CHAR(512)	First 512 bytes of the SQL statement text	First 512 bytes of the SQL statement
Note: This format is defined by member EZDAEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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

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.

28	1C	BINARY(4)	Requested function	The function being performed This field contains one of the following: • **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
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	• **0** - Connected to local RDB • **1** - Connected to remote RDB
94	5E	CHAR(1)	Commitment control level	• **’A’** - Commit *ALL • **’C’** - Commit *CHANGE • **’N’** - Commit *NONE • **’S’** - Commit *CS (cursor stability) • **’L’** - Commit *RR (repeatable read)
95	5F	CHAR(10)	Default SQL collection	Name of the default SQL collection used by the iSeries Database Server
105	69	CHAR(129)	Reserved	Reserved for future parameters
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
Note: This format is defined by member EZDAEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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	The function being performed This field contains one of the following: • **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

Offset				
Dec	Hex	Type	Field	Description
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
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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Exit point QIBM_QZDA_ROI1 format ZDAR0200

Offset				
Dec	Hex	Type	Field	Description
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: • **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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Data queue server:

Identify exit point for data queue server.

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				
Dec	**Hex**	**Type**	**Field**	**Description**
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	The function being performed • **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				
Dec	**Hex**	**Type**	**Field**	**Description**
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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Network print server:

Identify exit points for network print server.

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				
Dec	**Hex**	**Type**	**Field**	**Description**
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 QNPSERVR.
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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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 QNPSERVR
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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Central server:

Identify exit points for 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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: • **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: • **X'0000'** - Basic license information • **X'0001'** - Detailed license information
Note: This format is defined by member EZSCEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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: • **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 • **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, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC 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				
Dec	**Hex**	**Type**	**Field**	**Description**
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_NLS, the format name is ZSCN0100.
28	1C	BINARY(4)	Requested function	The function being performed • **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

Offset				
Dec	Hex	Type	Field	Description
40	28	BINARY(2)	Type of conversion	Requested mapping type: • **X'0001'** - Round trip • **X'0002'** - Substitution mapping • **X'0003'** - Best-fit mapping
Note: This format is defined by member EZSCEP in files H, QRPGSRC, QRPGLESRC, QLBLSRC and QCBLLESRC in library QSYSINC.				

Remote command and distributed program call server:

Identify exit point for remote command and 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				
Dec	Hex	Type	Field	Description
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 **X'1002'** - Remote command
32	20	CHAR(10)	Reserved	Not used for remote command requests
42	2A	CHAR(10)	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				
Dec	Hex	Type	Field	Description
0	0	CHAR(10)	User profile name	The name of the user profile that is calling the server

Offset				
Dec	**Hex**	**Type**	**Field**	**Description**
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.
56	38	CHAR(*)	Parameter information	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. • BINARY(4) - Length of parameter information for this parameter • BINARY(4) - Maximum length of parameter • BINARY(2) - Parameter usage type – **1** - Input – **2** - Output – **3** - Input / output • CHAR(*) - Parameter string

Signon server:

Identify 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				
Dec	Hex	Type	Field	Description
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.
28	1C	BINARY(4)	Requested function	The function being performed • **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

The sample exit programs in this topic 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: Create exit programs with RPG: The following example illustrates how to set up a user exit program with RPG*.

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

```
 **
   ** i5/OS 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 TFREQ
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 RSRSV
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 SPLF0l00
I                                       33  42 NPJOBN
I                                       43  52 NPUSRN
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 ZSCL0l00 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 ZSCS0l00 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 ZSCN0l00 for retrive 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* SPEClFIC PARAMETERS FOR DATABASE SERVER
I*
I                                       21  28 DBFMT
I                                    B  29  320DBFID
I*
```

```
     I* THE FOLLOWING PARAMETERS ADDITIONAL FOR FORMAT ZDAD0l00
     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 ZDAQ0l00
     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 ZDAR0l00
     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 ZSOY0l00 for TCP/IP signon server
     I*
     I                                       21  28 SOXFMT
     I                                    B  29  320SOFID
     I*
I***************************************************************
     I*
     I              '*VPRT     '          C         #VPRT
     I              '*TFRFCL   '          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              'QNPSERVR  '          C         #QNPSV
I              '*CNTRLSRV '          C         #CNTRL
I              '*RMTSRV   '          C         #RMTSV
I              '*SIGNON   '          C         #SIGN
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    RSQL
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                     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* 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*
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
```

Related information

"Code license and disclaimer information" on page 148

Examples: Create exit programs with 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.

```
/******************************************************************/
/*                                                                */
/* iSeries SERVERS- 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(&TFFMT) 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(&FSLEN) 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(&DBCRSR) 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(&DBFILR) 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 */
DCL VAR(&RCFMT) TYPE(*CHAR) LEN(8) /* Format name                */
DCL VAR(&RCFID) TYPE(*CHAR) LEN(4) /* Function identifier        */
DCL VAR(&RCPGM) TYPE(*CHAR) LEN(10) /* Program name               */
DCL VAR(&RCLIB) TYPE(*CHAR) LEN(10) /* Program library name       */
DCL VAR(&RCNUM) TYPE(*CHAR) LEN(4) /* Number of parms or cmdlen*/

DCL VAR(&RCDATA) TYPE(*CHAR) LEN(9999)/* Command string nor
parms */


/* SIGNON SERVER DECLARES */

DCL VAR(&SOFMT) TYPE(*CHAR) LEN(8) /* Format name
  */
DCL VAR(&SOFID) TYPE(*CHAR) LEN(4) /* Function identifier
  */

/***********************************/
/*                                 */
/* OTHER DECLARES                  */
/*                                 */
/**********************************/
 DCL VAR(&WRKLEN) TYPE(*CHAR) LEN(5)
 DCL VAR(&DECLEN) TYPE(*DEC) LEN(8 0)
/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*/                                                     */
/*                                            */
/* EXTRACT THE VARIOUS PARAMETERS FROM THE STRUCTURE  */
/*                                            */
/* * * * * * * * * * * * * * * * * * * * * * * * */

/* HEADER */
CHGVAR VAR(&USER)   VALUE(%SST(&REQUEST 1 10))
   CHGVAR VAR(&APPLIC) VALUE(%SST(&REQUEST 11 10))
   CHGVAR VAR(&FUNCTN) VALUE(%SST(&REQUEST 21 10))

/* VIRTUAL PRINTER */
   CHGVAR VAR(&VPOBJ)  VALUE(%SST(&REQUEST 31 10))
   CHGVAR VAR(&VPLIB)  VALUE(%SST(&REQUEST 41 10))
   CHGVAR VAR(&WRKLEN) VALUE(%SST(&REQUEST 71 5))
   CHGVAR VAR(&VPLEN)  VALUE(%BINARY(&WRKLEN 1 4))
   CHGVAR VAR(&VPOUTQ) VALUE(%SST(&REQUEST 76 10))
   CHGVAR VAR(&VPQLIB) VALUE(%SST(&REQUEST 86 10))


/* TRANSFER FUNCTION */
   CHGVAR VAR(&TFOBJ)  VALUE(%SST(&REQUEST 31 10))
   CHGVAR VAR(&TFLIB)  VALUE(%SST(&REQUEST 41 10))
   CHGVAR VAR(&TFMBR)  VALUE(%SST(&REQUEST 51 10))
   CHGVAR VAR(&TFFMT)  VALUE(%SST(&REQUEST 61 10))
   CHGVAR VAR(&WRKLEN) VALUE(%SST(&REQUEST 71 5))
   CHGVAR VAR(&TFLEN)  VALUE(%BINARY(&WRKLEN 1 4))
   CHGVAR VAR(&TFREQ)  VALUE(%SST(&REQUEST 76 1925))

/* FILE SERVER */
   CHGVAR VAR(&FSFID)      VALUE(%SST(&REQUEST   21   4))
   CHGVAR VAR(&FSFMT)      VALUE(%SST(&REQUEST   25   8))
   CHGVAR VAR(&FSREAD)     VALUE(%SST(&REQUEST   33   1))
   CHGVAR VAR(&FSWRITE)    VALUE(%SST(&REQUEST   34   1))
   CHGVAR VAR(&FSRDWRT)    VALUE(%SST(&REQUEST   35   1))
   CHGVAR VAR(&FSDLT)      VALUE(%SST(&REQUEST   36   1))
   CHGVAR VAR(&FSLEN)      VALUE(%SST(&REQUEST   37   4))
   CHGVAR VAR(&DECLEN)     VALUE(%BINARY(&FSLEN 1 4))
   CHGVAR VAR(&FSNAME)     VALUE(%SST(&REQUEST   41
&DECLEN))

/* DATA QUEUES */
   CHGVAR VAR(&DQQ)        VALUE(%SST(&REQUEST 31 10))
   CHGVAR VAR(&DQLIB)      VALUE(%SST(&REQUEST 41 10))
   CHGVAR VAR(&WRKLEN)     VALUE(%SST(&REQUEST 71  5))
   CHGVAR VAR(&DQLEN)      VALUE(%BINARY(&WRKLEN 1 4))
   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(&RSSTN)    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(&NPJOBN)   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(&DBCRSR)     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))

/************************************/
/*                                           */
/* BEGIN MAIN PROGRAM                        */
/*                                           */


 CHGVAR VAR(&STATUS) VALUE('1') /* INITIALIZE RETURN +
                          VALUE TO ACCEPT THE REQUEST */

 /* ADD LOGIC COMMON TO ALL SERVERS */

 /* PROCESS BASED ON SERVER ID */
 IF COND(&APPLIC *EQ '*VPRT') THEN(GOTO CMDLBL(VPRT))   /* IF VIRTUAL PRINTER */
 IF COND(&APPLIC *EQ '*TFRFCL') THEN(GOTO CMDLBL(TFR))  /* IF TRANSFER FUNCTIO*/
 IF COND(&APPLIC *EQ '*FILESRV') THEN(GOTO CMDLBL(FLR)) /* IF FILE SERVERS */
 IF COND(&APPLIC *EQ '*MSGFCL') THEN(GOTO CMDLBL(MSG))  /* IF MESSAGING FUNCT */
 IF COND(&APPLIC *EQ '*DQSRV') THEN(GOTO CMDLBL(DATAQ)) /* IF DATA QUEUES */
 IF COND(&APPLIC *EQ '*RQSRV') THEN(GOTO CMDLBL(RSQL))  /* IF REMOTE SQL */
 IF COND(&APPLIC *EQ '*SQL') THEN(GOTO CMDLBL(SQLINIT)) /* IF SQL */
 IF COND(&APPLIC *EQ '*NDB') THEN(GOTO CMDLBL(NDB))     /* IF NATIVE DATABASE */
 IF COND(&APPLIC *EQ '*SQLSRV') THEN(GOTO CMDLBL(SQLSRV)) /* IF SQL */
 IF COND(&APPLIC *EQ '*RTVOBJINF') THEN(GOTO CMDLBL(RTVOBJ)) /* IF RETRIEVE OB*/
 IF COND(&APPLIC *EQ '*DATAQSRV') THEN(GOTO CMDLBL(ODATAQ))  /* IF D*/
 IF COND(&APPLIC *EQ 'QNPSERVR') THEN(GOTO CMDLBL(NETPRT))  /* IF NETWORK PRI*/
 IF COND(&APPLIC *EQ '*CNTRLSRV') THEN(GOTO CMDLBL(CENTRAL)) /* IF CENTRAL SER*/
 IF COND(&APPLIC *EQ '*RMTSRV') THEN(GOTO CMDLBL(RMTCMD))    /* IF RMTCMD/DPC */
 IF COND(&APPLIC *EQ '*SIGNON') THEN(GOTO CMDLBL(SIGNON))  /* IF SIGNON */

 GOTO EXIT
/* * * * * * * * * * * * * * * * * * * * * * * */
/* SUBROUTINES                               */
/*                                           */
/* * * * * * * * * * * * * * * * * * * * * * * */

/* VIRTUAL PRlNTER */
 VPRT:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* TRANSFER FUNCTION */
```

```
 TFR:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* FILE SERVERS */
  FLR:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* MESSAGING FUNCTION */
  MSG:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* DATA QUEUES */
  DATAQ:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* REMOTE SQL */
  RSQL:

   /* SPECIFIC LOGIC GOES HERE */

   GOTO EXIT
/* 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
```

Related information

"Code license and disclaimer information" on page 148

iSeries NetServer administration

iSeries Access for Windows takes advantage of the IBM iSeries Support for Windows Network Neighborhood (iSeries NetServer). This function allows file serving and print serving.

For complete documentation on setting up, administering, and using the iSeries NetServer, see the iSeries NetServer information.

Related concepts

iSeries NetServer

Restrict users with policies and application administration

iSeries Access for Windows provides multiple methods of setting up restrictions and profiles. These include policies that can be set using Microsoft's policy editor, and the Application Administration function of iSeries Navigator.

iSeries Access for Windows supports two primary methods for implementing administrative control over your network: Application Administration and policies. Application Administration bases restrictions on the iSeries user profile, and is administered through iSeries Navigator. Policies mandate configuration settings and restrictions, and can apply to both specific PCs and individual Windows user profiles. As such, they offer greater granularity than Application Administration, but are significantly more difficult to set up and administer. In order to use policies, you must download the Microsoft System Policy Editor and configure your PCs and iSeries server for storage, retrieval, and application of policies you set. Generally, Application Administration is preferable if all of the functions you want to restrict are Application Administration-enabled, and if the version of the i5/OS server being used supports Application Administration.

For V5R2, Application Administration added support for Central Settings. The Central settings support in Application Administration provides the ability to manage most of the functions iSeries Access for Windows controls through the following policy templates:

- Runtime restrictions (caerestr.adm)
- Mandated connection properties (config.adm)
- Configuration policies (caecfg.adm)

For more information about Application Administration, refer to Application Administration.

To learn about policies, refer to the following topics:

Related concepts

"Microsoft System Policy Editor" on page 101
To create your own policy files, you need the Microsoft policy editor.

Application Administration

Overview of iSeries Access for Windows policies

Use iSeries Access for Windows System Policies to restrict users from certain actions, and to suggest or require certain configuration features.

System policies can apply to individual Windows user profiles, and specific PCs. However, these policies do not offer control over iSeries server resources, and are not a substitute for iSeries security. For a description of what you can do with these policies, refer to Types and scopes of policies.

Use of Group Policy to control use and configuration of iSeries Access for Windows had limited testing and can therefore provide unpredictable results. For additional information about Group Policy, see Microsoft documentation. The remainder of this topic discusses the tested, supported use of iSeries Access for Windows policies.

Policy support in your network

Policies can reside on a file server. When configured on a file server, each time users sign-on to their Windows workstation, their workstation downloads all the policies that apply to that Windows user profile. The user's PC applies the policies to the registry before the user does anything on the workstation. Each Windows operating system comes with the code needed to download policies.

To use the full capability of policies, you need the following:

- A primary logon server
- A policy server

You can use IBM iSeries Support for Windows Network Neighborhood (iSeries NetServer) as the policy server.

See Set up your system to use policies for more information.

Policy files

Policy definitions are contained in policy templates, which organize the policies into categories. iSeries Access for Windows provides five policy templates, one for each of the following functions:

- Restricting iSeries Access for Windows functions for a given system (sysname.adm)
- Restricting specific iSeries Access for Windows function at runtime (caerestr.adm)
- Restrict which components users may install or uninstall (caeinrst.adm)
- Mandate or suggest configuration settings for specific environments, the systems within those environments, and some configurable values for those systems (config.adm)
- Suggest or mandate global configurable values (caecfg.adm)

You must generate the policy templates with the CWBADGEN utility before creating or modifying specific policies. Then use the Microsoft System Policy Editor or the Microsoft Management Console Group Policy snap-in, gpedit.msc, to activate the templates and set their constituent policies. If using the Microsoft System Policy Editor, save the changes to a policy file. If using gpedit.msc, the policy settings are stored in a Group Policy Object automatically. See Microsoft documentation for details.

See Create policies for more information.

Related concepts

"Microsoft System Policy Editor" on page 101
To create your own policy files, you need the Microsoft policy editor.

"iSeries Access for Windows policy list" on page 103
iSeries Access for Windows supports Microsoft System Policies. Administrators can use policies to control which functions and settings are available to each user.

Related tasks

"Set up your system to use policies" on page 100
Download a policy file across the network.

"Create policy files" on page 101
Create or modify policies and store them in a policy file.

Types and scopes of policies

Each policy iSeries Access for Windows provides is either a restriction or a configuration policy, and can address one or more scopes.

Restriction policies

Restriction policies can usually be set to any scope and may have the following uses:

- Restrict or allow use of an iSeries Access for Windows function or action.
- Include restrictions for installing or uninstalling components, service packs, upgrades, or the entire product.
- Include several other restrictions. For example, you may restrict a certain type of data transfer upload, or you may restrict all types of data transfer uploads at once using the Prevent All Data Transfer to iSeries servers policy.
- Cause controls or options normally selectable to be hidden or ″greyed-out″.
- Notify the user when a restriction policy prevents a function they attempt from completing, usually by a message displayed in a console or a window.

Configuration policies

Configuration policies can only be set to a user scope, and may have the following uses:

- Pre-configure settings that the end user could normally configure themselves.
- Configure values, features that the user may normally enable or disable, lists of environments and connections.
- ″Grey-out″ a mandated value. When a configuration policy mandates a value, the input field for that value will not accept changes.

Configuration policies may be either suggested or mandated.

- Suggested: The value provided will be used unless explicitly configured by the user or set by an application program. This effectively overrides the normal default value iSeries Access for Windows would use, but does not force use of the value -- a new value may be specified, overriding the suggested value.
- Mandated: The value provided will be used -- neither the user nor application programs may change it.

Policy scopes

There are three scopes at which each policy may be set: machine scope, user scope and iSeries connection scope. Some policies may be set at more than one scope, while others may not.

Scope	Description
Machine scope	A policy set at this scope applies to all users of the PC. The only exception is when the same policy is set for a specific user to override the machine scope setting.
User Scope	A policy set at this scope can be applied on a per-user basis. It may be set for some users, but not others. It may be set for the ″Default User″ (any user without an individual policy configuration) as well. Some user scope policies provide a setting that allows a function regardless of the machine scope setting. When this setting is used, the machine scope setting is ignored.
iSeries Connection (or ″Per-System″) Scope	Some policies that can be set at user or machine scope may be more narrowly set at iSeries connection scope within the user or machine scope. When set at iSeries connection scope, the policy setting is applied only when working with the named iSeries system. For example, if a restriction policy is set at iSeries connection scope inside of user scope, where the iSeries system is named SYS1 and the user is USER1, the function is restricted only when USER1 works with SYS1. **Note:** If a policy is set at iSeries connection scope, this setting takes precedence over the user or machine scope setting. For example, if default user mode is mandated for user USER1 to be ″Use default user id″, but set for system SYS1 to be ″Use Windows user id and password″, when USER1 connects to SYS1, his Windows user id and password are used. When USER1 connects to any other system, the specified default user id is used **Note:** To enable setting policies at this scope, you must generate and use one or both of the following policy templates: • config.adm -- Configured environments and connections template • sysname.adm -- Per-system (by iSeries system name) template

Set up your system to use policies

Download a policy file across the network.

Complete the following steps to use iSeries Access for Windows policies by downloading a saved policy file across a network.

1. Configure an iSeries server for policies
2. Configure client PCs for policies
3. Create policy files

   Related concepts

   "Overview of iSeries Access for Windows policies" on page 98
   Use iSeries Access for Windows System Policies to restrict users from certain actions, and to suggest or require certain configuration features.

Configure an iSeries server for policies

Use the following steps to configure your iSeries server for serving policies. These steps assume that you have Windows PCs in your network.

- Configure your iSeries server as an iSeries NetServer, if this has not already been done.
- Create an integrated file system folder to hold your policy files.

  Related concepts

  iSeries NetServer

  Integrated file system

Configure client PCs for policies

Required configuration for client PCs to accept policy downloads from an iSeries system.

Note: This information applies to configuring PCs to download system policy files from a central location. You can also use iSeries Access for Windows policy support if the policies are stored locally or remotely, in a Group Policy Object (GPO). See Microsoft documentation for more information about group policy and Group Policy Objects.

Each Windows workstation in your network needs to download the policy file. You can download the `cwbpoluz` tool to do this for you. Download the tool from www.ibm.com/servers/eserver/iseries/access/cadownld.htm.

Alternatively, if you place the policy file on the **NETLOGON** share on the iSeries logon server, the user's PC will automatically download the policy file when the user logs onto an iSeries domain.

Related information

www.as400.ibm.com/clientaccess/cadownld.htm

Create policy files

Create or modify policies and store them in a policy file.

In order to create or modify specific policies and store them in a policy file, follow these steps:

1. Download the Microsoft System Policy Editor.
2. Create the policy templates for iSeries Access for Windows.
3. Create and update the policy file.

Note: A policy file is not needed if the Microsoft Management Console Group Policy snap-in, gpedit.msc, is used to set policies. See Microsoft documentation for more information.

Related concepts

"Overview of iSeries Access for Windows policies" on page 98
Use iSeries Access for Windows System Policies to restrict users from certain actions, and to suggest or require certain configuration features.

Microsoft System Policy Editor:

To create your own policy files, you need the Microsoft policy editor.

Use the Microsoft Web site to obtain the version of the policy editor that is supported on the Windows operating system that you are using. Search for **policy editor** at `www.microsoft.com`.

Follow the directions that come with the editor to extract the file and install the policy editor and templates.

Related concepts

"Restrict users with policies and application administration" on page 97
iSeries Access for Windows provides multiple methods of setting up restrictions and profiles. These include policies that can be set using Microsoft's policy editor, and the Application Administration function of iSeries Navigator.

"Overview of iSeries Access for Windows policies" on page 98
Use iSeries Access for Windows System Policies to restrict users from certain actions, and to suggest or require certain configuration features.

Related information

www.microsoft.com

Create policy templates for iSeries Access for Windows:

iSeries Access for Windows contains a program that creates the policy templates you need to control policies.

1. Open a command prompt window.
2. Go to the iSeries Access for Windows directory, normally located at:

   ```
   [C:]\Program Files\IBM\Client Access\
   ```
3. Type the command and parameter to give you the templates for the policies that you want to set.

Policy template commands

Command cwbadgen with parameters	Description
cwbadgen /ps S1034345 (Where s1034345 is the system name.)	Generates the template for setting system specific policies, S1034345.adm.
cwbadgen /std	Generates caecfg.adm (covers global configuration), caeinrst.adm (covers installation restrictions), & caerestr.adm (covers run time restrictions).
cwbadgen /cfg config.adm	Generates the config.adm (configuration policy based on system configurations that exist on the PC from which this command is run). Specify the name of the file after the /cfg argument. In this example the template file is config.adm.

Related concepts

"iSeries Access for Windows policy list" on page 103
iSeries Access for Windows supports Microsoft System Policies. Administrators can use policies to control which functions and settings are available to each user.

"Communication policy: Prevent connections to systems not previously defined" on page 110
Use this policy to prevent users from connecting to or configuring systems not yet defined.

"Policies by template" on page 143
Use these template files to control policies.

Create and update policy files:

Create policy files to control default computer or default user actions.

Note: The following instructions do not cover the use of Group Policy or the Microsoft Management Console Group Policy snap-in, although the instructions are similar. To administer iSeries Access for Windows functions using Group Policy, see the Microsoft documentation on Group Policy use.

1. Start the policy editor by double-clicking **poledit.exe**.
2. Go to **Options** → **Policy Template** → **Add**.
3. Go to the location where you stored the .adm files that you created in creating policy templates.
4. Select the .adm files that you want to add and click **Add**. Keep doing this until you have added all the .adm files that you want to use. Then click **OK**.
5. Go to **File** → **New Policy**.
6. Set your policies and save the policy file:

   ```
   \\QYOURSYS\POLICIES\ntconfig.pol
   ```

   Where:

   - QYOURSYS is the name of your iSeries NetServer.
   - POLICIES is the name of the shared file folder on your iSeries NetServer.
   - config.pol is the name of your policies file.

   To update the policy file, open your policy file with the policy editor, make your changes and save the file back to the above location.

Note: You must create and maintain individual policies for the different Windows operating systems. See Microsoft documentation for details.

iSeries Access for Windows policy list

iSeries Access for Windows supports Microsoft System Policies. Administrators can use policies to control which functions and settings are available to each user.

This topic lists all the policies that iSeries Access for Windows provides, and describes the effects and scope of each.

Sets of policies are defined by template files. You can generate policy templates (.adm files) for iSeries Access for Windows on a PC with iSeries Access for Windows installed using the **cwbadgen** command. See Create policy templates for iSeries Access for Windows for details. See a list of existing policies by selecting one of the following links:

- Policies by function

  Lists policies by the function they affect.
- Policies by template

  Lists the templates and their associated policies.

For a general description of policies in iSeries Access for Windows, see Overview of iSeries Access for Windows policies.

Related concepts

"Overview of iSeries Access for Windows policies" on page 98
Use iSeries Access for Windows System Policies to restrict users from certain actions, and to suggest or require certain configuration features.

Related tasks

"Create policy templates for iSeries Access for Windows" on page 101
iSeries Access for Windows contains a program that creates the policy templates you need to control policies.

Policies by function

Set these policies to control iSeries Access for Windows functions.

The following table lists iSeries Access for Windows policies by the function they affect.

Function	Related policies
.NET Data provider	Prevent .NET Data provider usage
ActiveX Automation Objects	• Prevent data transfer upload automation object • Prevent data transfer download automation object • Prevent remote command automation object • Prevent remote program automation object • Prevent data queue automation object

Function	Related policies
Communications	• Default user mode • TCP/IP Lookup • Port lookup mode • Require secure sockets • Prevent changes to active environment • Prevent changes to environment list • Prevent connections to systems not previously defined • Prevent use of non-mandated environments • Connection timeout
Data Transfer: Uploads	• Prevent all data transfer to an iSeries server • Prevent appending and replacing host files • Prevent Data Transfer GUI uploads • Prevent usage of RFROMPCB • Prevent autostart uploads
Data Transfer: Downloads	• Prevent all data transfer from an iSeries server • Prevent Data Transfer GUI downloads. • Prevent usage of RTOPCB • Prevent autostart downloads
Data Transfer: iSeries server file creation	• Prevent host file creation • Prevent Wizard iSeries server file creation • Prevent non-Wizard iSeries server file creation
Directory update	Prevent use of directory update
Incoming Remote Command	• Run as system • Command mode • Cache security • Allow generic security • Generic security runs command as logged on user
Install	• Selective Setup source directory • Prevent setup • Prevent selective setup • Prevent uninstall • Prevent check service pack level • Prevent installation of service pack • Prevent upgrades • Prevent installation of individual components
License management	Time to delay before license is released
National Language Support	• ANSI code page • OEM code page • EBCDIC code page • Bi-directional transformation of data
ODBC	• Named data sources • Prevent program generated data sources
OLE DB	Prevent OLE DB provider usage

Function	Related policies
iSeries Navigator	Prevent usage of iSeries Navigator
Passwords	• Warn user before iSeries password expires • Prevent iSeries Access for Windows password changes
PC5250 Emulation	• Prevent configuration of display sessions • Prevent configuration of printer sessions • Prevent usage of PC5250 emulator • Maximum number of PC5250 Sessions • Prevent changing of .WS profiles • Prevent menu configuration • Prevent toolbar configuration • Prevent multi-session configuration • Prevent keyboard configuration • Prevent mouse configuration • Prevent Java applet execution • Prevent access to macros • Prevent profile imports in Emulator Session Manager • Prevent profile deletion in Emulator Session Manager • Prevent directory changes in Emulator Session Manager
PC Commands	• Cwblogon • Cwbcfg • Cwbback • Cwbrest • Cwbenv • cwbundbs • Wrksplf • wrkmsg • wrkprt • wrkusrj
Service	• When to check • Delay time • Frequency • Copy image to PC • Run silently • Service path • Autostart background service job
User Interface	Prevent creation of desktop icons

Policies by function: .NET Data provider:

Control .NET provider by policies.

.NET Data Provider policy: Prevent .NET Data Provider usage:

Use this policy to prevent use of the iSeries Access for Windows .NET Data Provider. When not restricted by this policy, the .NET Data Provider allows applications using Microsoft ′s .NET framework to access DB2 UDB for iSeries Databases.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Policies by function: ActiveX automation objects:

Control ActiveX by policies.

ActiveX policy: Prevent data transfer upload automation object:

Use this policy to prevent use of the data transfer upload automation object.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		

ActiveX policy: Prevent data transfer download automation object:

Use this policy to prevent users from using the data transfer download automation object.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		

ActiveX policy: Prevent Remote Command automation object:

Use this policy to prevent use of the Remote Command automation object.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		

ActiveX policy: Prevent Remote program automation object:

Use this policy to prevent use of the Remote program automation object.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		

ActiveX policy: Prevent data queue automation object:

Use this policy to prevent users from using the data queue automation object.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		

Policies by function: Communication:

Control communication functions by policies.

Communication policy: Default user mode:

Use this policy to configure the default user mode when connecting to an iSeries server.

You can configure the default user mode to:

- Always prompt for user ID and password.
- Use a default user ID, which you must specify with this policy.
- Use the Windows user ID and password of the logged-on user.
- Use the Kerberos principal name, no prompting.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		X

Communications policy: TCP/IP Address Lookup Mode:

Use this policy to suggest or mandate how frequently iSeries IP addresses should be looked up.

You can use this policy to set the TCP/IP address lookup mode to:

- Lookup always (do not cache the address)
- Lookup once per hour
- Lookup once per day
- Lookup once per week
- Lookup after Windows has been re-started
- Never look it up

Note: If you select Never look it up, you must also specify an IP address to use.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		X

Communication policy: Port lookup mode:

Use this policy to suggest or mandate the method used, and location to look in, to obtain the TCP/IP port number for a specific server program on the iSeries server.

A per-system (iSeries connection scope) mandate will always override a global (machine scope) mandate, or a user-configured value, for port lookup mode.

You can use this policy to set the port lookup mode to:

- Lookup locally
- Lookup on server
- Use standard port

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (may override machine setting)	Per iSeries connection
	X		X

Communication policy: Require Secure Sockets:

Use this policy to require that a user connect to the iSeries server using the secure sockets layer (SSL).

To use this policy, SSL must be installed and configured on both the iSeries server and the client PC. It is not possible to mandate that SSL be turned off. It is always possible for a user to elect to use SSL, assuming that it is installed and configured on both the iSeries server and the client PC.

If this policy mandates the use of SSL, any connection attempt that cannot use SSL will fail. This means that if the user does not have SSL installed, or if the iSeries system is incapable of using SSL or does not have the SSL-capable versions of the host servers started, no connections to iSeries servers can be made!

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		X

Communication policy: Prevent changes to active environment:

Use this policy to prevent switching the active environment. Use it to force users to use a specific environment.

If there is no active environment specified, or if the active environment is set to an invalid value, iSeries Access for Windows uses the ″My iSeries Connections″ environment. If that environment doesn’t exist, iSeries Access for Windows uses the first environment in the list of environments.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate

Policy Type		
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Communication policy: Prevent environment list changes:

Use this policy to prevent a user, or users of a PC, from making changes to the list of connection environments. Specifically, the user will not be able to add new environments, rename existing environments, or delete existing environments.

This policy only prevents manipulation of the environment list. The user will still be permitted to manipulate the contents of an environment, i.e. add/rename/remove systems in the environment.

This policy will be of interest to administrators who want to tightly control which iSeries servers their iSeries Access for Windows users can connect to.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Communication policy: Prevent connections to systems not previously defined:

Use this policy to prevent users from connecting to or configuring systems not yet defined.

This policy does not mandate systems or environments. Mandating these is done by creating and using the policy template config.adm. See Create policy templates for iSeries Access for Windows to read about how to do this.

When this policy is used:

- Systems not yet defined may not be used for any iSeries Access for Windows function.
- New systems may not be defined.
- Systems may still be deleted, but cannot then be re-defined.
- Environments may still be added, deleted, or renamed.

When environments and systems are mandated:

- Systems not yet defined may be used for iSeries Access for Windows functions.
- New systems and environments may be defined.
- Systems and environments already defined may not be deleted.

To force a user to use, and not modify, a set of environments and systems, use this policy along with mandating environments and systems.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related tasks

"Create policy templates for iSeries Access for Windows" on page 101
iSeries Access for Windows contains a program that creates the policy templates you need to control policies.

Communication policy: Prevent use of non-mandated environments:

Use this policy to restrict users to using only connection environments mandated by the administrator. This policy is helpful for administrators who want to tightly control which iSeries servers can be accessed by their users.

To mandate use of a collection of environments, and systems within those environments, create a policy template using cwbadgen.exe and the /cfg option. Then include this template when building the policy file. The creation of this template should be done only when the environments and systems configured on the PC are exactly those the users should use.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Communication policy: Timeout value:

Use this policy to enforce a timeout value. However, the user can overwrite the policy programmatically, or by manually configuring the value for the specific system being connected to.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Policies by function: Data Transfer:

Control Data Transfer functions by policies.

Policies by function: Data Transfer uploads:

Control Data Transfer upload functions by policies.

Data Transfer policy: Prevent all Data Transfer to iSeries server:

Use this policy to prevent uploading data to an iSeries server with Data Transfer.

Using this policy is equivalent to using all of the following policies:
- Prevent appending and replacing host files
- Prevent Data Transfer GUI uploads
- Prevent usage of RFROMPCB
- Prevent autostart uploads

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent appending and replacing host files:

Use this policy to prevent users from using Data Transfer to append or replace an existing file on the iSeries server.

This restriction is also set when you use the more general policy Prevent all Data Transfer Uploads.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection

Policy Scope			
X	X	X	X

Data Transfer policy: Prevent Data Transfer GUI upload:

Use this policy to prevent users form uploading data to an iSeries server with the Data Transfer GUI.

Using the more general policy Prevent all Data Transfer uploads also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent usage of RFROMPCB:

Use this policy to prevent use of the RFROMPCB command line program.

The more general policy Prevent all Data Transfer uploads also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent autostart uploads:

Use this policy to restrict a user or a PC from running Data Transfer autostart requests to send data to an iSeries server.

The more general policy, Prevent all data transfer uploads to an iSeries server also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Data Transfer iSeries server file creation:

Control creation of a server file by policies.

- Prevent host file creation
- Prevent Wizard iSeries server file creation
- Prevent non-Wizard iSeries server file creation

Related concepts

"Data Transfer policy: Prevent Wizard iSeries server file creation"
Use this policy to prevent users from creating iSeries server files with the Data Transfer Wizard.

"Data Transfer policy: Prevent non-Wizard iSeries server file creation" on page 115
Use this policy to prevent users from creating iSeries server files with the non-Wizard version of Data Transfer.

Data Transfer policy: Prevent host file creation:

Use this policy to prevent the creation of iSeries host server files by using Data Transfer.

Setting this policy is equivalent to using these policies:

- Prevent Wizard iSeries server file creation.
- Prevent non-wizard iSeries server file creation.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Related concepts

"Data Transfer policy: Prevent Wizard iSeries server file creation"
Use this policy to prevent users from creating iSeries server files with the Data Transfer Wizard.

"Data Transfer policy: Prevent non-Wizard iSeries server file creation" on page 115
Use this policy to prevent users from creating iSeries server files with the non-Wizard version of Data Transfer.

Data Transfer policy: Prevent Wizard iSeries server file creation:

Use this policy to prevent users from creating iSeries server files with the Data Transfer Wizard.

Using the more general policy Prevent host file creation also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Related concepts

"Data Transfer policy: Data Transfer iSeries server file creation" on page 114
Control creation of a server file by policies.

"Data Transfer policy: Prevent host file creation" on page 114
Use this policy to prevent the creation of iSeries host server files by using Data Transfer.

Data Transfer policy: Prevent non-Wizard iSeries server file creation:

Use this policy to prevent users from creating iSeries server files with the non-Wizard version of Data Transfer.

Using the more general policy prevent host file creation also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Related concepts

"Data Transfer policy: Data Transfer iSeries server file creation" on page 114
Control creation of a server file by policies.

"Data Transfer policy: Prevent host file creation" on page 114
Use this policy to prevent the creation of iSeries host server files by using Data Transfer.

Data Transfer policy: Data Transfer downloads:

Control Data Transfer downloads by policies.

Data Transfer policy: Prevent all Data Transfers from an iSeries server:

Use this policy to prevent downloading data from an iSeries server with Data Transfer.

Using this policy is equivalent to using all of the following policies:

- Prevent Data Transfer GUI download

- Prevent usage of RTOPCB
- Prevent autostart download

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent Data Transfer GUI download:

Use this policy to prevent users form downloading data from an iSeries server with the Data Transfer GUI.

Using the more general policy Prevent all Data Transfer Downloads also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent usage of RTOPCB:

Use this policy to prevent the use of the RTOPCB command line program.

The more general policy Prevent all Data Transfer downloads also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Data Transfer policy: Prevent autostart downloads:

Use this policy to restrict a user or a PC from running Data Transfer autostart requests to download data from an iSeries server.

The more general policy, Prevent all data transfer downloads from an iSeries server also sets this restriction.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Policies by function: Directory update:

Control directory update by a policy.

Directory update policy: Prevent using directory update:

Use this policy to prevent usage of the Directory Update function.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by function: Incoming Remote Command:

Control Incoming Remote Command function by policies.

Incoming Remote Command policy: Run as system:

Use this policy to prevent the use of the **Run as system** option for Incoming Remote Command.

For more information, see the **Incoming Remote Command** tab of the **iSeries Access for Windows Properties** interface.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Incoming Remote Command policy: Command mode:

Use this policy to prevent the use of the **Command mode** option for Incoming Remote Command.

For more information, see the **Incoming Remote Command** tab of the **iSeries Access for Windows Properties** interface.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Incoming Remote Command policy: Cache security:

Use this policy to prevent the use of the **Cache security** option for Incoming Remote Command.

For more information, see the **Incoming Remote Command** tab of the **iSeries Access for Windows Properties** interface.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Incoming Remote Command policy: Allow generic security:

Use this policy to prevent the use of the **Allow generic security** option for Incoming Remote Command.

For more information, see the **Incoming Remote Command** tab of the **iSeries Access for Windows Properties** interface.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Incoming Remote Command policy: Generic Security Runs command as logged on user:

Use this policy to prevent the use of the Generic Security Runs Command As Logged On User option for Incoming Remote Command. For more information, refer to the online help.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Policies by function: Installation:

Control installation function by policies.

Installation policy: Selective setup source directory:

Used to mandate the path from which components may be installed using Selective Setup.

The path from which iSeries Access for Windows was originally installed is stored in the iSeries Access for Windows configuration at installation time, and is normally the path used by Selective Setup. Because a path is configured, using this policy simply to suggest a different path will have no effect, since configured values override suggested ones. However, a mandated path will override the configured path, as expected.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection

Policy Scope			
X			

Installation policy: Prevent Setup:

Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

Note: Other types of installation, such as installing a new release over an old one (upgrading), are not prevented. Other policies exist for controlling the other types of installation actions; these are the following:

- Prevent uninstall
- Prevent installation of service pack
- Prevent upgrades
- Prevent selective setup
- Prevent installation of individual components

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent uninstall" on page 121
Use this policy to prevent users from uninstalling iSeries Access for Windows.

"Installation policy: Prevent installation of service pack" on page 122
Use this policy to prevent the installing of an iSeries Access for Windows service pack.

"Installation policy: Prevent upgrades" on page 122
Use this policy to prevent installing a newer release of iSeries Access for Windows over an older one, or re-installing the same release.

"Installation policy: Prevent Selective Setup"
Use this policy to prevent the use of iSeries Access for Windows Selective Setup, so that once iSeries Access for Windows is installed, no additional iSeries Access for Windows components may be installed later on the PC.

"Installation policy: Prevent installation of individual components" on page 123
Use these policies to prevent installation of individual components or subcomponents of iSeries Access for Windows.

Installation policy: Prevent Selective Setup:

Use this policy to prevent the use of iSeries Access for Windows Selective Setup, so that once iSeries Access for Windows is installed, no additional iSeries Access for Windows components may be installed later on the PC.

To restrict only certain components from being installed by Selective Setup, use the individually-installable component policies.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent Setup" on page 120
Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

"Installation policy: Prevent installation of individual components" on page 123
Use these policies to prevent installation of individual components or subcomponents of iSeries Access for Windows.

Installation policy: Prevent uninstall:

Use this policy to prevent users from uninstalling iSeries Access for Windows.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent Setup" on page 120
Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

Installation policy: Prevent check service pack level:

Use this policy to prevent the running of the iSeries Access for Windows Check Service Level utility.

This program normally runs at a certain time after Windows start-up, or may not run at all, according to how the user has configured iSeries Access for Windows. (Configuration for this program is found within iSeries Access for Windows Properties in the Windows Control Panel, on the Service tab.) It may also be run manually by the user at any time. If this policy is enabled, Check Service Level may not be run either automatically or manually.

If the policy for preventing service pack installation is set, you may want to prevent checking of the service pack level as well. If you do not, when the check runs it might display a message stating that a service pack is available to install, even though the user cannot install it.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent installation of service pack"
Use this policy to prevent the installing of an iSeries Access for Windows service pack.

Installation policy: Prevent installation of service pack:

Use this policy to prevent the installing of an iSeries Access for Windows service pack.

Note that no other installation restrictions prevent the installation of a service pack.

If you set this policy, you may want to set the Prevent Check Service Pack Level policy as well. If you do not, the check may result in a message being displayed to the user leading them to believe that they can install an iSeries Access for Windows service pack, when they really cannot.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent Setup" on page 120
Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

"Installation policy: Prevent check service pack level" on page 121
Use this policy to prevent the running of the iSeries Access for Windows Check Service Level utility.

Installation policy: Prevent upgrades:

Use this policy to prevent installing a newer release of iSeries Access for Windows over an older one, or re-installing the same release.

This policy will not prevent you from installing iSeries Access for Windows on a PC that has never had iSeries Access for Windows installed, or on a PC from which it has been uninstalled. To prevent installations on PCs that don't have any version of iSeries Access for Windows, use the Prevent Setup policy.

PolicyType		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"Installation policy: Prevent Setup" on page 120
Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

Installation policy: Prevent installation of individual components:

Use these policies to prevent installation of individual components or subcomponents of iSeries Access for Windows.

Normally, these components can be installed during an initial installation of iSeries Access for Windows, an upgrade to a newer release, or when using Selective Setup after the product has been installed. When policy is used to prevent the installation of a specific component, none of those methods may be used to install that component. In fact, the restricted component will not even appear as an installation choice.

Once a component is installed, using this policy will not cause the component to be uninstalled. If, however, the component is uninstalled later, it cannot be re-installed due to the policy restriction. One case in which this occurs is during an upgrade from one release to another. The first release is uninstalled, then when the new release is installed, policy-restricted components cannot be installed again.

Some of the components are each made up of multiple subcomponents. In these cases, one policy usually exists to restrict installation of the whole component, while other policies exist to allow preventing the installation of subcomponents within the higher level component.

The list of all individual components and subcomponents whose installation may be restricted by policy is as follows:

Individual component	**Sub-component**
Base component	• On-line User's Guide • Incoming Remote Command • Directory update

Individual component	**Sub-component**
iSeries Navigator	• Basic operations • Work Management • System configuration • Network • Security • Users and groups • Database • File systems • Backup • Management Central – Commands – Packages and Products – Monitors • Application Administration • Logical Systems • Advanced Function Presentation™
Unknown iSeries Navigator Plug-ins	
Data Access	• Data Transfer – Data Transfer installation options • OLE DB Provider • .NET Data Provider • ODBC • Lotus 1-2-3 file format support
AFP Workbench viewer	
IBM Toolbox for Java	
PC5250 display and printer emulation and subcomponents	
Printer drivers	• AFP printer driver • SCS printer driver
Operations Console	
Application Development Toolkit	
EZ-Setup	

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

“Installation policy: Prevent Setup” on page 120
Use this policy to prevent running the iSeries Access for Windows Setup program to install iSeries Access for Windows on a PC on which it is not currently installed.

“Installation policy: Prevent Selective Setup” on page 120
Use this policy to prevent the use of iSeries Access for Windows Selective Setup, so that once iSeries Access for Windows is installed, no additional iSeries Access for Windows components may be installed later on the PC.

Policies by function: License management:

Control license manage by policies.

You can use these policies to control the amount of time to delay before license is released.

Related concepts

“License policy: Time to delay before releasing iSeries Access for Windows license”
Use this policy to control how long iSeries Access for Windows waits to give up an iSeries Access for Windows license after all licensed programs have ended.

License policy: Time to delay before releasing iSeries Access for Windows license:

Use this policy to control how long iSeries Access for Windows waits to give up an iSeries Access for Windows license after all licensed programs have ended.

This setting is normally configurable by the user on the Other tab of iSeries Access for Windows Properties. The value this policy may be set to is the number of minutes iSeries Access for Windows should wait. If no value is set by policy, and the user has not configured a value, the default is to wait 10 minutes before giving up the license.

Even though the policy setting allows only minutes to be specified, the value on the iSeries Access for Windows Properties Other tab is shown in both hours and minutes.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Related concepts

“Policies by function: License management”
Control license manage by policies.

Policies by function: National Language Support:

Control National Language Support function by policies.

National Language Support policy: ANSI code page:

Use this policy to control which ANSI code page should be used for specific users when using iSeries Access for Windows functions.

This setting is normally configured on the Language tab of iSeries Access for Windows Properties. If no value is set using this policy, and no value has been configured by the user, the PC's default ANSI code page will be used.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

National Language Support policy: OEM code page:

Use this policy to control which OEM code page should be used when using iSeries Access for Windows functions.

This setting is normally configured on the Language tab of iSeries Access for Windows Properties. If no value is set using this policy, and no value has been configured by the user, the PC's default OEM code page will be used.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

National Language Support policy: EBCDIC code page:

Use this policy to control which EBCDIC CCSID should be used by iSeries Access for Windows functions.

This setting is normally configured on the Language tab of iSeries Access for Windows Properties. If no value is set using this policy, and no value has been configured by the user, the EBCDIC CCSID is taken from the iSeries job serving the client.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Language policy: BiDi Transform:

Suggests or mandates the value for the BiDi Transform setting on the iSeries Access for Windows Control Panel.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Policies by function: ODBC:

Control ODBC functions by policies.

ODBC policy: Prevent use of named data sources:

Use this policy to restrict the use of named data sources when using iSeries Access for Windows ODBC support.

A ″named data source″ is one that:
- has been created by the user or a program and given a specific name, and
- is specified using the **DSN** option when connecting.

A user may create a named data source using the iSeries Access for Windows ODBC Administration program. A program may create a named data source too -- by calling, for example, SQLCreateDataSource.

A program may create an ODBC connection by calling SQLDriverConnect. If the DSN option is used, it specifies a named data source to use. If the FILEDSN option is used, it specifies the name of a file that contains connection options. The file name is not a data source name, hence use of FILEDSN is not use of a named data source.

The restriction options for this policy are the following:
- **Allow all:** All named data sources may be used.
- **Allow listed sources:** Only those sources specifically listed in this policy may be used. To view or change the list, click the Show button.
- **Prevent using named data sources:** No named data sources may be used.

If when connecting no named data source is specified, the data source used will be a temporary one, called a ″program generated data source.″ The use of program generated data sources can be restricted using the Prevent use of program generated data sources policy.

This policy is an override of **machine setting enabled**.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		X

Related concepts

"ODBC policy: Prevent use of program generated data sources"
Use this policy to restrict the use of program generated data sources when using iSeries Access for Windows ODBC support.

ODBC policy: Prevent use of program generated data sources:

Use this policy to restrict the use of program generated data sources when using iSeries Access for Windows ODBC support.

A ″program generated data source″ is one that is created temporarily when an ODBC connection is made without using the DSN option to specify the name of the data source. Note that use of the FILEDSN option does not mean the data source used is named. FILEDSN simply specifies the name of a file containing connection options, not the name of a data source.

If a program first creates a data source (using SQLCreateDataSource, for example) and then connects using the DSN option, the data source is not considered a program generated data source, but a named data source. To restrict the use of named data sources, use the Prevent use of named data sources policy.

This policy is an override of **machine setting enabled**.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X		X

Related concepts

"ODBC policy: Prevent use of named data sources" on page 127
Use this policy to restrict the use of named data sources when using iSeries Access for Windows ODBC support.

Policies by function: OLE DB:

Control usage of the OLE DB provider by policies.

OLE DB Provider policy: Prevent OLE DB Provider usage:

Use this policy to prevent use of the iSeries Access for Windows OLE DB providers.

When not restricted by this policy, the OLE DB Provider is used to access iSeries database files, stored procedures, data queues, CL commands, and programs.

Note: A single policy covers all OLE DB providers so, if this prevent policy is set, none of the OLE DB providers will work.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	X

Policies by function: iSeries Navigator:

Control usage of iSeries Navigator by policies.

iSeries Navigator policy: Prevent usage of iSeries Navigator:

Use this policy to prevent the use of iSeries Navigator.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by function: Passwords:

Control passwords by policies.

Password policy: Warn user before iSeries password expires:

Use this policy to control if and when iSeries Access for Windows will warn a user whose iSeries password is near expiration.

If the policy is set, the number of days before expiration at which point the user is to be warned must be specified as well. Normally these can be configured by the user using the Passwords tab of iSeries Access for Windows Properties. If no value is set by policy and the user has not configured a value, the default action is to warn the user when a password is within 14 days of expiring.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Password policy: Prevent iSeries Access for Windows password changes:

Use this policy to prevent PC users from changing their iSeries server passwords through the Passwords tab of the iSeries Access for Windows Properties.

This policy can not prevent users from changing their iSeries server passwords when using a PC5250 emulation session.

Note: If this policy is not in effect, the user may still be prevented from changing his iSeries server password by restrictions placed on his account by the iSeries system administrator.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by function: PC5250 emulation:

Control PC5250 functions by policies.

PC5250 emulation policy: Prevent configuration of display sessions:

Use this policy to prevent configuration of new PC5250 emulator display sessions.

The settings of display sessions you have already configured can be viewed, but not changed. This policy does not control the use of display sessions, only the configuring of new ones.

This policy does not prevent configuration of new PC5250 printer sessions. To prevent such configuration, use the Prevent configuration of printer session policy.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"PC5250 emulation policy: Prevent configuration of printer sessions"
Use this policy to prevent configuration of new PC5250 emulator printer sessions.

PC5250 emulation policy: Prevent configuration of printer sessions:

Use this policy to prevent configuration of new PC5250 emulator printer sessions.

The settings of printer sessions you have already configured can be viewed, but not changed. This policy does not control the use of printer sessions, only the configuring of new ones.

This policy does not prevent configuration of new PC5250 display sessions. To prevent such configuration, use the Prevent configuration of display sessions policy.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Related concepts

"PC5250 emulation policy: Prevent configuration of display sessions" on page 130
Use this policy to prevent configuration of new PC5250 emulator display sessions.

PC5250 emulation policy: Prevent usage of PC5250 Emulator:

Use this policy to prevent use of the PC5250 emulator.

When you set this policy, display and printer sessions are both unavailable.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Maximum number of PC5250 sessions:

Use this policy to restrict connecting to a specific iSeries system using PC5250 emulation.

Users for whom this policy is set may only connect up to the specified number of PC5250 emulation sessions to the specified iSeries at once. Both display and printer sessions are included in this maximum count.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		X

PC5250 emulation policy: Prevent changing of .WS profiles:

Use this policy to control a user's ability to change configuration information pertaining to communication.

This includes emulator configuration (the **communication** → **Configure menu item**).

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent menu configuration:

Use this policy to control the user's ability to read and change configuration information pertaining to the menu.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate

Policy Type		
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent toolbar configuration:

Use this policy to control the user's ability to read and change configuration information pertaining to the toolbar.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent multi-session configuration:

Use this policy to control a user's ability to read, execute and control information pertaining to multiple sessions.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent keyboard configuration:

Use this policy to control the user's ability to read and change configuration information pertaining to the keyboard.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate

Policy Type		
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent mouse configuration:

Use this policy to control the user's ability to read and change configuration information pertaining to the mouse.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent Java applet execution:

Use this policy to control the user's ability to execute Personal Communications 5250 Java applets via the **Actions** → **Run Java Applet** menu item.

Note: PC5250 as included with iSeries Access for Windows does not support the **Actions** → **Run Java Applet** interface.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent access to macros:

Use this policy to control the user's ability to record or play macros.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent profile imports in Emulator Session Manager:

Use this policy to control the user's ability to import emulator profiles in the Emulator Session Manager.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent profile deletion in Emulator Session Manager:

Use this policy to control the user's ability to delete emulator profiles in the Emulator Session Manager.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC5250 emulation policy: Prevent directory changes in Emulator Session Manager:

Use this policy to control the user's ability to change the Emulator Session Manager directory.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by function: PC Commands:

Restrict use of PC commands by policies.

PC command policy: Prevent use of Cwblogon.exe:

Use this policy to prevent use of the Cwblogon utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of Cwbcfg.exe:

Use this policy to prevent use of the Cwbcfg utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of Cwbback.exe:

Use this policy to prevent use of the cwbback utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of Cwbrest.exe:

Use this policy to prevent use of the Cwbrest utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of Cwbenv.exe:

Use this policy to prevent use of the Cwbenv utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of cwbundbs.exe:

Use this policy to prevent use of the cwbundbs utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of Wrksplf.exe:

Use this policy to prevent use of the Wrksplf utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of wrkmsg.exe:

Use this policy to prevent use of the wrkmsg utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of wrkprt.exe:

Use this policy to prevent use of the wrkprt utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

PC command policy: Prevent use of wrkusrj.exe:

Use this policy to prevent use of the wrkusrj utility.

For more information about this PC command, refer to the iSeries Access for Windows online User's Guide.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by function: Service:

Control service by policies.

Service policy: When to check service level:

Use this policy to control when iSeries Access for Windows Check Service Level should run.

This setting is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. The setting choices for the policy are the same as those in Client Access Properties. If no value is set by policy, and the user has not configured a value, the default is Periodically, and the Frequency setting dictates how many days to wait between checks.

If you set this policy, you may want to set the Delay Time policy and the Frequency policy as well. Depending on the setting of the When To Check policy, these policies may also have an effect.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Related concepts

"Service policy: Delay time"
Use this policy to control how long iSeries Access for Windows will wait, after Windows starts, to automatically run the Check Service Level utility.

"Service policy: Frequency" on page 141
Use this policy to control how frequently iSeries Access for Windows Check Service Level will run.

Service policy: Delay time:

Use this policy to control how long iSeries Access for Windows will wait, after Windows starts, to automatically run the Check Service Level utility.

This setting is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. Note that this setting has no effect when the When To Check setting is Never, since Check Service Level will never be run automatically in that case.

The value this policy may be set to is the number of seconds iSeries Access for Windows should wait. If no value is set by policy, and the user has not configured a value, the default is to wait 60 seconds before Check Service Level runs.

Note that even though the policy setting allows a number of seconds to be specified, the value on the iSeries Access for Windows Properties Service tab is shown in minutes. It is the nearest number of whole minutes in the number of seconds specified in the policy.

If you set this policy, you may want to set the When to check policy and the Frequency policy as well.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Related concepts

"Service policy: When to check service level" on page 139
Use this policy to control when iSeries Access for Windows Check Service Level should run.

"Service policy: Frequency"
Use this policy to control how frequently iSeries Access for Windows Check Service Level will run.

Service policy: Frequency:

Use this policy to control how frequently iSeries Access for Windows Check Service Level will run.

This setting is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. If no value is set by policy, and the user has not configured a value, the default is to check once every 28 days. Note that this policy will have no effect unless the When To Check value is set to Periodically. If you set this policy, you may want to set the When To Check policy and the Delay time policy as well.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Related concepts

"Service policy: When to check service level" on page 139
Use this policy to control when iSeries Access for Windows Check Service Level should run.

"Service policy: Delay time" on page 140
Use this policy to control how long iSeries Access for Windows will wait, after Windows starts, to automatically run the Check Service Level utility.

Service policy: Copy image to PC:

Use this policy to control whether or not iSeries Access for Windows installation functions copy the installation image files to the PC before starting the install.

This value is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. If no value is set by policy, and the user has not configured a value, the default is to not copy the installation image to the PC.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection

Policy Scope			
X			

Service policy: Run silently:

Use this policy to control whether updates and release upgrades to iSeries Access for Windows software run silently -- that is, with no user interaction.

This value is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. If no value is set by policy, and the user has not configured a value, the default is that such updates and upgrades will run interactively.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Service policy: Service path:

Use this policy to set the location at which iSeries Access for Windows will look for upgrades and service packs when checking levels and installing.

This value is normally configurable by the user on the Service tab of iSeries Access for Windows Properties. If no value is set by policy, and the user has not configured a value, the default is the location iSeries Access for Windows was last installed from.

Note: iSeries Access for Windows configures this value to be the initial installation path during installation. Since configured values are always used before checking for suggested values, suggesting a value using this policy will have no effect.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X			

Service policy: Autostart background service job:

Use this policy to control whether the background service job starts automatically at Windows start-up time.

This is normally configured on the Service tab of iSeries Access for Windows Properties. If no value is set using this property, and no value has been configured by the user, the background service job is not started automatically.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
	X	X

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
	X		

Policies by function: User interface:

Control user interface by policies.

User interface policy: Prevent creation of desktop icons:

Use this policy to prevent creation of iSeries Access for Windows system desktop icons.

These icons serve as a direct way to open and connect a specific application, such as iSeries Navigator or a user-defined program, to a specific iSeries system. They can normally be created by right-clicking on the iSeries system name in iSeries Navigator and selecting Create Desktop Icon. They may also be created by right-clicking on the Windows desktop, then selecting New, and iSeries Desktop Icon.

While this policy can restrict creation of that type of icon, other types of desktop icons may still be created using iSeries Navigator. These include the following:
- Copies of printer output files.
- Shortcuts to iSeries Navigator folders, such as Messages.
- Files or folders from the iSeries Integrated File System.

There are no iSeries Access for Windows policies that restrict creation of such icons.

Policy Type		
Restriction	Configuration	
	Suggestion	Mandate
X		

Policy Scope			
Per PC (all users)	Per user	Per user setting (May override machine setting)	Per iSeries connection
X	X	X	

Policies by template:

Use these template files to control policies.

Choose from the following templates. See Create policy templates for iSeries Access for Windows for more information.

Related tasks

"Create policy templates for iSeries Access for Windows" on page 101
iSeries Access for Windows contains a program that creates the policy templates you need to control policies.

Caecfg.adm:

Use these policies to suggest or mandate specific iSeries Access for Windows configurable values.

Function	Policies
Communications	• Default user mode • TCP/IP address lookup • Port lookup mode • Require secure sockets • Connection timeout • Active Environment
Passwords	• Warn users before iSeries password expires
Incoming Remote Command	• Run as system • Command mode • Cache security • Allow generic security • Generic security runs as logged on user
National language support	• ANSI code page • OEM code page • EBCDIC code page • Enable BiDi transformation of data
Service	• When to check • Delay time • Frequency • Copy image to PC • Run silently • Service path • Autostart background service job
Install	Selective setup source directory
License management	Time to delay before iSeries Access for Windows license is released

Caerestr.adm: iSeries Access for Windows Runtime Restrictions:

Use these policies to restrict specific iSeries Access for Windows functions.

Function	Related policies
.NET Data provider	Prevent .NET Data provider usage

Function	Related policies
ActiveX Automation Objects	• Prevent data transfer upload automation object • Prevent data transfer download automation object • Prevent remote command automation object • Prevent remote program automation object • Prevent data queue automation object
Data Transfer: Uploads	• Prevent all data transfer to an iSeries server • Prevent appending and replacing host files • Prevent Data Transfer GUI uploads • Prevent usage of RFROMPCB • Prevent autostart uploads
Data Transfer: Downloads	• Prevent all data transfer from an iSeries server • Prevent Data Transfer GUI downloads • Prevent usage of RTOPCB • Prevent autostart downloads
Data Transfer: iSeries server file creation	• Prevent host file creation • Prevent Wizard iSeries server file creation • Prevent non-Wizard iSeries server file creation
Directory update	Prevent using directory update
Passwords	Prevent iSeries Access for Windows password changes
iSeries Navigator	Prevent use of iSeries Navigator
Communications	• Prevent changes to active environment • Prevent changes to active environment list • Prevent connections to systems not previously defined • Prevent use of non-mandated environments
ODBC	• Named data sources • Prevent program generated data sources
OLE DB provider	Prevent OLE DB provider usage
PC5250 emulation	• Prevent configuration of display sessions • Prevent configuration of printer sessions • Prevent usage of PC5250 emulator • Maximum number of PC5250 Sessions • Prevent changing of .WS profiles • Prevent menu configuration • Prevent toolbar configuration • Prevent multi-session configuration • Prevent keyboard configuration • Prevent mouse configuration • Prevent Java applet execution • Prevent access to macros • Prevent profile imports in Emulator Session Manager • Prevent profile deletion in Emulator Session Manager • Prevent directory changes in Emulator Session Manager

Function	Related policies
PC commands	• Cwblogon • Cwbcfg • Cwbback • Cwbrest • Cwbenv • cwbundbs • Wrksplf • wrkmsg • wrkprt • wrkusrj
User interface	Prevent creation of desktop icons

Config.adm: iSeries Access for Windows mandated connections:

Use these policies to mandate configuration settings for specific environments, the systems within those environments, and some configurable values for those systems.

This template only stores the environments and systems that are configured on your PC when you generate the template. If you want to add or remove environments and systems from the template, re-run cwbadgen with the /cfg option. Using the /cfg option also lets you specify a filename for the configuration template. This allows you to keep several different versions of the file, reflecting various configurations.

Note: Mandated systems will not appear in iSeries Navigator unless you specify at least one of the policies listed for that system.

Function	Related policies
Environment1: system1: Communications	• Default user mode • TCP/IP Lookup • Port lookup mode • Require secure sockets
Environment1: system2:	
Environronent2: system1:	

Caeinrst.adm: Install restrictions:

Use these policies to restrict which items users may install or uninstall, as well as other functions related to installation.

Function	Related policies
Installation	• Prevent setup • Prevent selective setup • Prevent uninstall • Prevent check service pack level • Prevent installation of service pack • Prevent upgrades • Prevent installation of individual components

SYSNAME.adm: Per-system policies:

Use these policies to restrict specific iSeries Access for Windows functions for a given system.

Function	Related policies
Data Transfer: Upload	• Prevent all data transfer to an iSeries server • Prevent appending and replacing host files • Prevent Data Transfer GUI upload • Prevent usage of RFROMPCB • Prevent autostart upload
Data Transfer: Downloads	• Prevent all data transfer from an iSeries server • Prevent Data Transfer GUI downloads. • Prevent usage of RTOPCB • Prevent autostart downloads
Data Transfer: iSeries server file creation	• Prevent host file creation • Prevent Wizard iSeries server file creation • Prevent non-Wizard iSeries server file creation
ODBC	• Named data sources • Prevent program generated data sources
OLE DB provider	Prevent OLE DB provider usage
.NET Data provider	Prevent .NET Data provider usage
PC5250 emulation	Maximum number of PC5250 Sessions

Secure Sockets Layer administration

Secure Sockets Layer (SSL) is a popular security scheme that allows the PC client to authenticate the server and encrypts all data and requests.

Use SSL when transferring sensitive data between clients and servers. The transfer of credit card and bank statement information are examples of client/server transactions that typically take advantage of SSL. There is an increased cost in performance with SSL because of the added encryption and decryption processing.

iSeries Access for Windows includes optionally-installable support for Secure Sockets Layer (SSL) and a way to manage key databases with **IBM Key Management**. All functions of iSeries Access for Windows can communicate over SSL except Incoming Remote Command. iSeries Access for Windows allows SSL communications with the iSeries server at the 128-bit, or higher, level of encryption.

Note:

- Client authentication is available for PC5250.
- Both 32-bit and 64-bit SSL support are installed on the client, when the SSL component is installed on a 64-bit Windows operating system.

To configure SSL, see the topic collection at **Networking** → **Networking security** → **Secure Sockets Layer (SSL)** .

Related concepts

Secure Sockets Layer (SSL)

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.

Related concepts

"iSeries Access for Windows: Administration," on page 1
Use this topic to administer iSeries Access for Windows in your client/server environment.

"Examples: Create exit programs with RPG" on page 83

"Examples: Create exit programs with CL commands" on page 89

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This iSeries Access publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of iSeries Access.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

1-2-3
Advanced Function Presentation
AFP
DB2
DB2 Universal Database
Distributed Relational Database Architecture
DRDA
i5/OS
IBM
IBM (logo)
iSeries
Lotus
NetServer
OS/2

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation 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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Printed in USA