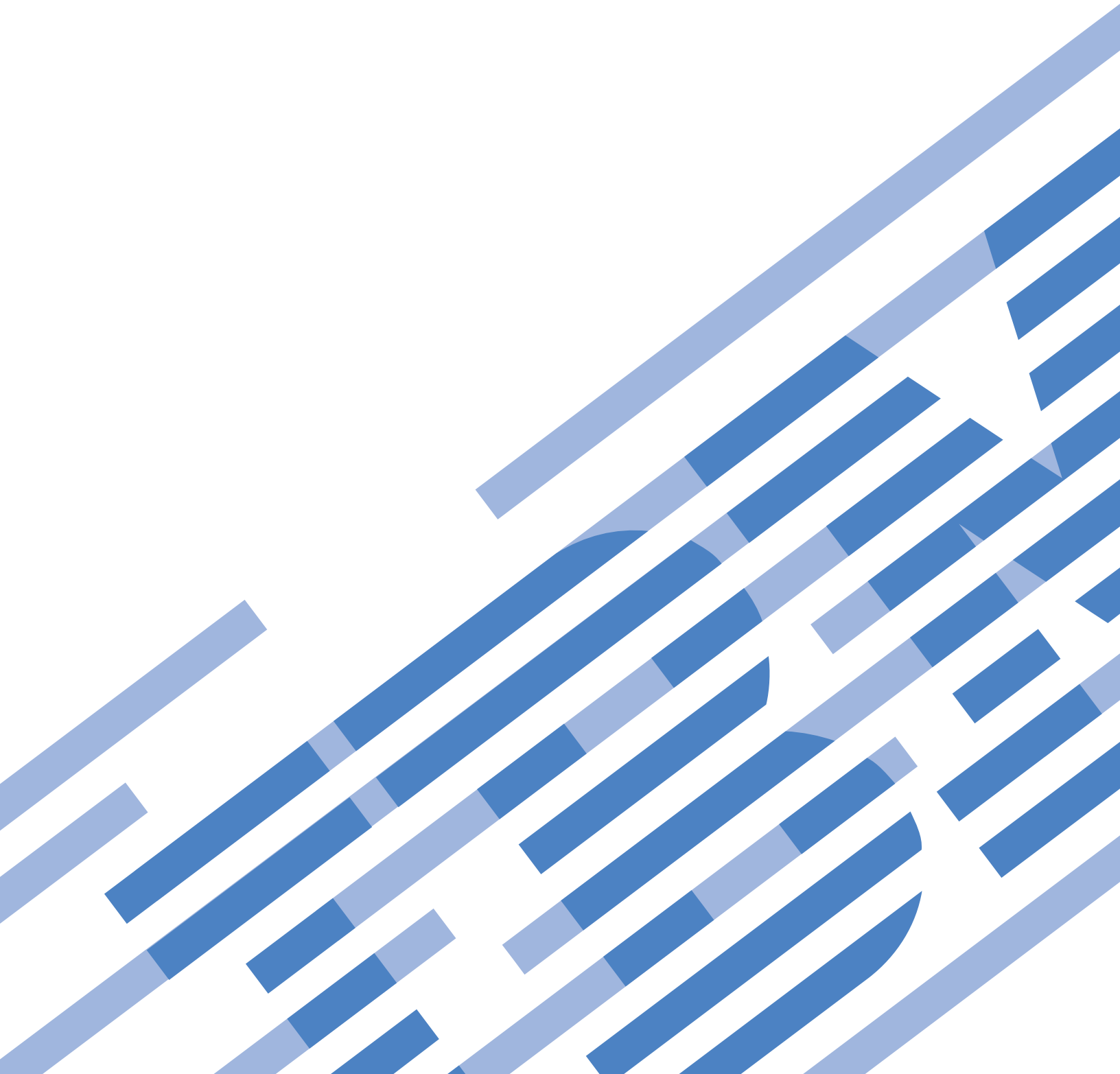
IBM Systems - iSeries

Server Support APIs

Version 5 Release 4



IBM Systems - iSeries

Server Support APIs

Version 5 Release 4

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 123.

Sixth Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Server Support APIs

The Server Support APIs include:

- "iSeries Support for Windows Network Neighborhood Server APIs"
- "Integrated Operating Environments APIs" on page 53
- "Dynamic Host Configuration Protocol Server Exit Programs" on page 116
- "Host Server Exit Programs" on page 117

Top | APIs by category

APIs

These are the APIs for this category.

iSeries Support for Windows Network Neighborhood Server APIs

The iSeries Support for Windows Network Neighborhood Server (iSeries NetServer) APIs allow personal computers to access file and print resources on the iSeries(TM) by using the networking support provided with their operating systems. Operating systems that are supported by iSeries NetServer include Windows(TM) 2000, Windows XP, Windows Server 2003, and Linux/Samba.

For more information, see the iSeries NetServer topic.

The iSeries NetServer APIs and their functions are:

- "Add File Server Share (QZLSADFS) API" on page 2 (QZLSADFS) shares a portion of the integrated file system with the network using the iSeries Support for Windows Network Neighborhood server.
- "Add Print Server Share (QZLSADPS) API" on page 5 (QZLSADPS) shares an output queue with the network by using the iSeries Support for Windows Network Neighborhood server.
- "Change File Server Share (QZLSCHFS) API" on page 7 (QZLSCHFS) changes the information about a shared portion of the integrated file system.
- "Change Print Server Share (QZLSCHPS) API" on page 10 (QZLSCHPS) changes the information about a shared printer.
- "Change Server Guest (QZLSCHSG) API" on page 12 (QZLSCHSG) changes the user profile used by the iSeries Support for Windows Network Neighborhood server when an unknown user requests access to shared resources on the system.
- "Change Server Information (QZLSCHSI) API" on page 13 (QZLSCHSI) changes general configuration information for the iSeries Support for Windows Network Neighborhood server.
- "Change Server Name (QZLSCHSN) API" on page 17 (QZLSCHSN) changes the name by which the iSeries Support for Windows Network Neighborhood server is known on the network.
- "End Server (QZLSENDS) API" on page 18 (QZLSENDS) ends the jobs that support the iSeries Support for Windows Network Neighborhood server.
- "End Server Session (QZLSENSS) API" on page 19 (QZLSENSS) ends a single session or all the sessions that were established from the specified workstation.
- "List Server Information (QZLSLSTI) API" on page 20 (QZLSLSTI) lists information about the server for configuration, connections, sessions, shares, or statistical information.
- "Open List of Server Information (QZLSOLST) API" on page 36 (QZLSOLST) opens a list of information about the server for share, configuration, session, connection, or statistical information.

- "Remove Server Share (QZLSRMS) API" on page 51 (QZLSRMS) removes a share from the list of files or printers on the server.
- "Start Server (QZLSSTRS) API" on page 52 (QZLSSTRS) starts the jobs necessary for the iSeries Support for Windows Network Neighborhood server to run.

Top | "Server Support APIs," on page 1 | APIs by category

Add File Server Share (QZLSADFS) API

Required Parameter Group:

1	Share name	Input	Char(12)
2	Path name	Input	Char(*)
3	Length of path name	Input	Binary(4)
4	CCSID encoding of path name	Input	Binary(4)
5	Text description	Input	Char(50)
6	Permissions	Input	Binary(4)
7	Maximum users	Input	Binary(4)
8	Error code	I/O	Char(*)

Optional Parameter Group:

9	CCSID used for text conversion	Input	Binary(4)
10	Enable text conversion	Input	Char(1)
11	File extension table	Input	Char(*)
12	Number of table entries	Input	Binary(4)

Default Public Authority: *USE
Threadsafe: No

The Add File Server Share (QZLSADFS) API shares a portion of the integrated file system with the network using the iSeries Support for Windows Network Neighborhood server.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority or own the integrated file system directory.

Required Parameter Group

Share name
INPUT; CHAR(12)

The name of the share to be added.

Path name
INPUT; CHAR(*)

The path in the integrated file system to be shared with the network. A forward slash, '/', is required as the first character.

Length of path name
INPUT; BINARY(4)

The length, in bytes, of the path name. The maximum length is 1024 bytes.

CCSID encoding of path name
INPUT; BINARY(4)

The coded character set ID (CCSID) of the path name and the file extension table entries. 0 will result in the current job CCSID being used.

Text description
INPUT; CHAR(50)

The text description for this share.

Permissions
INPUT; BINARY(4)

The access available from the network for this share.

1 Read only
2 Read/Write

Maximum users
INPUT; BINARY(4)

The maximum number of users who can concurrently access this share.

This value must be greater than or equal to zero, or -1 (X'FFFFFFFF'), which means there is no limit to the number of users who can concurrently use this share.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter Group

CCSID used for text conversion
INPUT; BINARY(4)

The client CCSID that is used for text file data conversion. Text file data conversion is performed using this CCSID and the current CCSID of the system file.

A value of 0 indicates that the user would like to use the currently configured CCSID for the server. See the Get Related Default CCSID (CDRGRDC) API for additional information on related CCSIDs.

If the CCSID used for text conversion parameter is omitted, the API assumes a default value of 0.

Enable text conversion
INPUT; CHAR(1)

Whether the server enables text conversion for this share. Possible values are:

0 Text conversion not enabled
1 Text conversion enabled
2 Text conversion enabled and mixed data is allowed

If the enable text conversion parameter is omitted, the API assumes a default value of 0.

File extension table
INPUT; CHAR(*)

An array of table entries. There is no limit to the number of values specified. For the layout of a table entry, see "Format of File Extension Table Entry" on page 4.

If the file extension table parameter is omitted, the API assumes that there are no file extensions for text conversion. If the file extension table parameter is specified, the number of table entries parameter must be specified also.

Number of table entries
INPUT; BINARY(4)

The number of table entries.

If the number of table entries parameter is left out, the API assumes a default value of 0. If the number of table entries parameter is specified, the file extension table parameter must be specified also.

Format of File Extension Table Entry

The following shows the format of a file extension table entry. For detailed descriptions of the fields in the table, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of file extension
4	4	CHAR(46)	File extension

Field Descriptions

File extension. A file extension string that is null-terminated. Possible values include:

*	The server will convert all files.
.	The server will convert all files without an extension.
TXT, .TXT	The server will convert all files ending with .TXT (that is, a.TXT, a.b.c.TXT).
..TXT, ...TXT, ...TXT	Extensions with more than one leading period will have no effect on the server. No translation will be done.
*T*T*	The server will convert all files ending with an extension that substitutes any number of characters for the * wild card (that is, a.T123T, b.TXT, c.TEST).
T?T	The server will convert all files ending with an extension that substitutes any one character for the ? wild card (that is, a.T1T, b.TXT).

Length of file extension. The size in bytes of the file extension. The length does not include the byte used for null-termination.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68A E	Error occurred while working with shared resource &2.
CPFB68B E	Character is not valid for value &3.
CPFB68D E	Length specified in parameter &2 for API &1 not valid.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Add Print Server Share (QZLSADPS) API

Required Parameter Group:

1	Share name	Input	Char(12)
2	Qualified output queue name	Input	Char(20)
3	Text description	Input	Char(50)
4	Spooled file type	Input	Binary(4)
5	Print driver type	Input	Char(50)
6	Error code	I/O	Char(*)

Optional Parameter Group 1:

7	Qualified printer file	Input	Char(20)

Optional Parameter Group 2:

8	Publish print share	Input	Char(1)

Default Public Authority: *USE
Threadsafe: No

The Add Print Server Share (QZLSADPS) API shares a system output queue with the network by using the iSeries Support for Windows Network Neighborhood server.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority or own the system output queue.

Required Parameter Group

Share name
INPUT; CHAR(12)

The name of the share to be added.

Qualified output queue name
INPUT; CHAR(20)

The output queue. The first 10 characters contain the name of the output queue. The second 10 characters contain the name of the library that contains the output queue.

Text description
INPUT; CHAR(50)

The text description for this share.

Spooled file type
INPUT; BINARY(4)

The type of spooled files that are created using this share. This is based on spooled file data expected to be sent by the client; no conversion will be done.

Note: In most cases, the automatic type sense value is selected. The other options are needed for downward compatibility. Values can be:

1 User ASCII (*USERASCII)
2 Advanced Function Printing (*AFP)
3 SNA character string (*SCS)
4 Automatic type sensing

Print driver type
INPUT; CHAR(50)

The text string that identifies the print driver that is appropriate for this share. When personal computers connect to this shared printer, this identifies the print driver that they should use. This text should match the name of a print driver known to the personal computer operating system. For example, 'IBM AFP 3820' is an Advanced Function Print driver type.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter Group 1

Qualified printer file
INPUT; CHAR(20)

A template containing attributes used to create spooled files. The first 10 characters contain the name of the printer file. The second 10 characters contain the name of the library that contains the printer file.

If the printer file parameter is left out, the API assumes that no printer file will be used to create spooled files.

This parameter must be specified when optional parameter 2 is specified. Blanks can be specified for the printer file to indicate no printer file will be used to create spooled files.

Optional Parameter Group 2

Publish print share
INPUT; CHAR(1)

Whether to publish this print share. Possible values are:

0 Do not publish the print share
1 Publish the print share

If the publish print share parameter is omitted, the API assumes a default value of 0.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68A E	Error occurred while working with shared resource &2.
CPFB68B E	Character is not valid for value &3.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Change File Server Share (QZLSCHFS) API

Required Parameter Group:

1	Share name	Input	Char(12)
2	Path name	Input	Char(*)
3	Length of path name	Input	Binary(4)
4	CCSID encoding of path name	Input	Binary(4)
5	Text description	Input	Char(50)
6	Permissions	Input	Binary(4)
7	Maximum users	Input	Binary(4)
8	Error code	I/O	Char(*)

Optional Parameter Group:

9	CCSID used for text conversion	Input	Binary(4)
10	Enable text conversion	Input	Char(1)
11	File extension table	Input	Char(*)
12	Number of table entries	Input	Binary(4)

Default Public Authority: *USE
Threadsafe: No

The Change File Server Share (QZLSCHFS) API changes the information about a shared portion of the integrated file system.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority or own the integrated file system directory.

Required Parameter Group

Share name
INPUT; CHAR(12)

The name of the share to be changed.

Path name
INPUT; CHAR(*)

The path in the integrated file system to be shared with the network. A forward slash, '/', is required as the first character.

Length of path name
INPUT; BINARY(4)

The length, in bytes, of the path name. The maximum length is 1024 bytes.

CCSID encoding of path name
INPUT; BINARY(4)

The coded character set ID (CCSID) of the path name and the file extension table entries. 0 will result in the current job CCSID being used.

Text description
INPUT; CHAR(50)

The text description for this share.

Permissions
INPUT; BINARY(4)

The access available from the network for this share. Possible values follow.

1 Read only
2 Read/Write

Maximum users
INPUT; BINARY(4)

The maximum number of users who can concurrently access this share.

This value must be greater than or equal to zero, or -1 (X'FFFFFFFF'), which means there is no limit to the number of users who can concurrently use this share.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter Group

CCSID used for text conversion
INPUT; BINARY(4)

The client CCSID that is used for text file data conversion. Text file data conversion is performed using this CCSID and the current CCSID of the system file.

A value of 0 indicates that the user would like to use the currently configured CCSID for the server. See the Get Related Default CCSID (CDRGRDC) API for additional information on related CCSIDs.

If the CCSID used for text conversion parameter is omitted, the API assumes that no change is requested and the current value remains unchanged.

Enable text conversion
INPUT; CHAR(1)

Whether the server enables text conversion for this share. Possible values are:

0 Text conversion not enabled
1 Text conversion enabled
2 Text conversion enabled and mixed data is allowed

If the enable text conversion parameter is omitted, the API assumes that no change is requested and the current value remains unchanged.

File extension table
INPUT; CHAR(*)

An array of table entries. There is no limit to the number of values specified. For the layout of a table entry, see "Format of File Extension Table Entry" on page 9.

If the file extension table parameter is omitted, the API assumes that no change is requested and current values remain unchanged. If the file extension table parameter is specified, the number of table entries parameter must be specified also.

Number of table entries
INPUT; BINARY(4)

The number of table entries.

If the number of table entries parameter is left out, the API assumes that no change is requested and the current value remains unchanged. If the number of table entries parameter is specified, the file extension table parameter must be specified also.

Format of File Extension Table Entry

The following table shows the format of a file extension table entry. For detailed descriptions of the fields in the table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of file extension
4	4	CHAR(46)	File extension

Field Descriptions

File extension. A file extension string that is null-terminated. Possible values include:

*	The server will convert all files.
.	The server will convert all files without an extension.
TXT, .TXT	The server will convert anything ending in .TXT (that is, a.TXT, a.b.c.TXT).
..TXT, ...TXT, ...TXT	Extensions with more than one leading period will have no effect on the server. No translation will be done.
*T*T*	The server will convert all files ending with an extension that substitutes any number of characters for the * wild card (that is, a.T123T, b.TXT, c.TEST).
T?T	The server will convert all files ending with an extension that substitutes any one character for the ? wild card (that is, a.T1T, b.TXT).

Length of file extension. The size in bytes of the file extension. The length does not include the byte used for null-termination.

Error Messages

Message ID	**Error Message Text**
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68A E	Error occurred while working with shared resource &2.
CPFB68B E	Character is not valid for value &3.
CPFB68D E	Length specified in parameter &2 for API &1 not valid.
CPFB693 E	Data conversion failed for &5 API.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Change Print Server Share (QZLSCHPS) API

Required Parameter Group:

1	Share name	Input	Char(12)
2	Qualified output queue name	Input	Char(20)
3	Text description	Input	Char(50)
4	Spooled file type	Input	Binary(4)
5	Print driver type	Input	Char(50)
6	Error code	I/O	Char(*)

Optional Parameter Group 1:

7	Qualified printer file	Input	Char(20)

Optional Parameter Group 2:

8	Publish print share	Input	Char(1)

Default Public Authority: *USE
Threadsafe: No

The Change Print Server Share (QZLSCHPS) API changes the information about a shared printer.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority or own the system output queue.

Required Parameter Group

Share name
INPUT; CHAR(12)

The name of the share to be changed.

Qualified output queue name
INPUT; CHAR(20)

The name of the qualified output queue. The first 10 characters identify the name of the output queue. The second 10 characters contain the name of the library that contains the output queue.

Text description
INPUT; CHAR(50)

The text description for this share.

Spooled file type
INPUT; BINARY(4)

The type of spooled files that are created by using this share. This is based on spooled file data expected to be sent by the client; no conversion will be done.

Note: In most cases, the automatic type sense value is selected. The other options are needed for downward compatibility. Values can be:

1 User ASCII (*USERASCII)
2 Advanced Function Printing (*AFP)
3 SNA character string (*SCS)
4 Automatic type sensing

Print driver type
INPUT; CHAR(50)

The text string that identifies the print driver appropriate for this share. When personal computers connect to this shared printer, this identifies the print driver that they should use. This text should match the name of a print driver known to the personal computer operating system. For example, 'IBM AFP 3820' is an Advanced Function Print driver type.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter Group 1

Qualified printer file
INPUT; CHAR(20)

A template containing attributes used to create spooled files. The first 10 characters contain the name of the printer file. The second 10 characters contain the name of the library that contains the printer file. If the printer file parameter is omitted, the API assumes that no change is requested and the current value remains unchanged.

This parameter must be specified when optional parameter 2 is specified.

Optional Parameter Group 2

Publish print share
INPUT; CHAR(1)

Whether to publish this print share. Possible values are:

0 Do not publish the print share
1 Publish the print share

If the publish print share parameter is omitted, the API assumes no change is requested and the current value remains unchanged.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68A E	Error occurred while working with shared resource &2.
CPFB68B E	Character is not valid for value &3.
CPFB693 E	Data conversion failed for &5 API.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Change Server Guest (QZLSCHSG) API

Required Parameter Group:

1	Guest user profile	Input	Char(10)
2	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Change Server Guest (QZLSCHSG) API changes the user profile used by the iSeries Support for Windows Network Neighborhood server when an unknown user requests access to shared resources on the system. The changes made with this API do not take effect until the server is restarted.

A user profile with any special authority cannot be specified as the guest profile.

Authorities and Locks

To use this API, you must have both *IOSYSCFG and *SECADM special authorities.

Guest user profile
*USE

Required Parameter Group

Guest user profile
INPUT; CHAR(10)

The user profile for the iSeries Support for Windows Network Neighborhood server access by unknown users. If this field is blank, unknown users will not have access to resources on the system.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB686 E	Error configuring iSeries Support for Windows Network Neighborhood (iSeries NetServer).
CPFB687 E	Cannot find the guest user profile &3.
CPFB68C E	Requested guest user profile &2 has incorrect authority.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Change Server Information (QZLSCHSI) API

Required Parameter Group:

1	Request variable	Input	Char(*)
2	Length of request variable	Input	Binary(4)
3	Format	Input	Char(8)
4	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Change Server Information (QZLSCHSI) API changes general configuration information for the iSeries Support for Windows Network Neighborhood (iSeries NetServer) server. Changes made with this API using format ZLSS0200 take effect immediately. All other changes made with this API using format ZLSS0100 do not take effect until the server is restarted.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority. To use format ZLSS0200 to enable an iSeries NetServer user, you must have *SECADM special authority and *OBJMGT and *USE authority to the system user profile.

Required Parameter Group

Request variable
INPUT; CHAR(*)

A variable that is used to pass values for the information to be changed.

For more information, see "Format of Information Changed."

Length of request variable
INPUT; BINARY(4)

The length of the variable that is used to pass information to be changed. For Format ZLSS0100, if this length is less than the total length of the format, the fields not completely included by the length use their default values. For Format ZLSS0200, if this length does not include all of the user names to be enabled, only the user names completely included by the length are enabled.

Format
INPUT; CHAR(8)

The content and format of the information to be changed by the simplified server. The following format names must be used:

ZLSS0100	Server information
ZLSS0200	Enable user information

For more information, see "Format of Information Changed."

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Information Changed

The following table shows how the content of the request variable is organized. For descriptions of each field, see "Field Descriptions" on page 14.

Format ZLSS0100

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	CCSID
4	4	BINARY(4)	Idle time-out
8	8	BINARY(4)	Opportunistic lock time-out
12	C	BINARY(4)	Browsing interval
16	10	CHAR(15)	WINS primary address
31	1F	CHAR(15)	WINS secondary address
46	2E	CHAR(224)	Scope ID
270	10E	CHAR(1)	WINS proxy
271	10F	CHAR(5)	Reserved
276	114	BINARY(4)	Server role
280	118	CHAR(1)	Authentication method
281	119	CHAR(3)	Reserved
284	11C	BINARY(4)	Message authentication
288	120	BINARY(4)	Minimum message severity
292	124	BINARY(4)	LAN Manager authentication

Format ZLSS0200

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Number of iSeries NetServer users to enable
This field repeats for each user.		CHAR(10)	iSeries NetServer user name

Field Descriptions

Authentication method. The authentication method used to authenticate users. Valid values are:

character 0 (X'F0') The server authenticates with encrypted passwords only.
character 1 (X'F1') The server authenticates with network authentication only.
character 2 (X'F2') The server authenticates with network authentication when possible, but it allows clients to use encrypted passwords when needed.

The default value for this field is character 0.

Note: Authentication method should only be set to character 1 in networks containing only Windows 2000 and newer iSeries NetServer clients that are configured to participate in a network authentication realm. All other Windows clients will be unable to connect to iSeries NetServer when network authentication only support is enabled.

Browsing interval. The amount of time, in seconds, between each server announcement that is used for browsing. The maximum allowable value is 720000 milliseconds (or twelve minutes). The minimum allowable value is zero, which means there will be no server announcements.

The default value for this field is 720000 milliseconds.

CCSID. The coded character set ID (CCSID) that is used for all clients connected to the server.

The default value for this field is the associated ASCII CCSID for the CCSID of the job used to start the server. See the Get Related Default CCSID (CDRGRDC) API for additional information on related CCSIDs.

Note: A value of 0 indicates that the user would like to use the associated ASCII CCSID for the CCSID of the job used to start the server. The field in the corresponding header file for this format may indicate code page, but the value to be entered as input must be a CCSID.

Idle time-out. The amount of time, in seconds, that a connection to the simplified server will remain active once activity has ceased on that connection. The idle time-out value must be greater than zero, or -1 (*X'FFFFFFFF'*) for no autodisconnect.

The default value for this field is 6000 seconds.

iSeries NetServer user name. The iSeries NetServer user name to be enabled.

LAN Manager (LANMAN) authentication. The level of restriction on the use of the LANMAN password hash for authentication. Valid values are:

0	LANMAN password hash is ignored if a stronger password hash is provided by the client.
1	LANMAN password hash is used only if a stronger password hash provided by the client does not match or if a stronger password hash is not provided.

The default value for this field is 0.

Message authentication. The status of message authentication. Valid values are:

0	Server does not support message authentication.
1	Server supports message authentication. Message authentication is negotiated between the client and the server.
2	Server requires message authentication for all connections.

The default value for this field is 0.

Minimum message severity. The minimum message severity of administrative alerts to send to users of the server. Valid values are:

-1	Administrative alert messages are not sent.
0-99	Administrative alert messages with a severity greater than or equal to the value are sent.

The default value for this field is -1.

Number of iSeries NetServer users to enable. The number of iSeries user names in the list to be enabled.

Opportunistic lock time-out. The amount of time, in seconds, that the server will wait for a response to a break lock request sent to a lock holder before forcefully removing the lock. Valid values are:

-1	Opportunistic locking is disabled.
1-600	The number of seconds to wait for a response to a break lock request.

The default value for this field is 30.

Scope ID. A string that names the network scope to be used by the Windows Internet Name Service (WINS) server.

The default value is nothing.

Server role The logon server role for the system. Valid values are:

0	The system is not a server
1	The system is a server

The default value for this field is 0.

WINS primary address. The IP address of the primary Windows Internet Name Service (WINS) server.

If this field is empty, it indicates no WINS primary server. This is the default.

WINS proxy. Enable or disable the system to act as a WINS proxy. This enables non-WINS clients to obtain name resolution from WINS. WINS is a dynamic naming service that resolves NetBIOS computer names to IP addresses. Valid values are:

character 0 (X'F0')	Disable WINS proxy
character 1 (X'F1')	Enable WINS proxy

The default value for this field is character 0.

WINS secondary address. The IP address of the secondary Windows Internet Name Service (WINS) server.

If this field is empty, it indicates no WINS secondary server. This is the default.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.

Message ID	Error Message Text
CPFB684 E	User does not have the correct authority for API &1.
CPFB686 E	Error configuring iSeries Support for Windows Network Neighborhood (iSeries NetServer).
CPFB690 E	Error changing iSeries Support for Windows Network Neighborhood (iSeries NetServer) information.
CPFB694 E	When enabling users, &1 users were not enabled.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Change Server Name (QZLSCHSN) API

Required Parameter Group:

1	Server name	Input	Char(15)
2	Domain name	Input	Char(15)
3	Text description	Input	Char(50)
4	Error code	I/O	Char(*)

Optional Parameter:

5	Allow system name	Input	Char(1)

Default Public Authority: *USE
Threadsafe: No

The Change Server Name (QZLSCHSN) API changes the name by which the iSeries Support for Windows Network Neighborhood server is known on the network. The changes made with this API do not take effect until the server is restarted.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority.

Required Parameter Group

Server name

INPUT; CHAR(15)

The name by which the system is known in the Windows Network Neighborhood.

Domain name

INPUT; CHAR(15)

The domain name (or workgroup) in which the server appears.

Text description

INPUT; CHAR(50)

The description of the system that appears on the network.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter

Allow system name
INPUT; CHAR(1)

Whether the server allows clients to connect using the system name found in the iSeries network attributes. Possible values are:

0 Do not allow access to the server using the TCP/IP system name.
1 Allow access to the server using the TCP/IP system name.

If the allow system name parameter is omitted, the API assumes that no change is requested and the current value remains unchanged.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C3A E	Value for parameter &2 for API &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB686 E	Error configuring iSeries Support for Windows Network Neighborhood (iSeries NetServer).
CPFB68B E	Character is not valid for value &3.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

End Server (QZLSENDS) API

Required Parameter:

1	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The End Server (QZLSENDS) API ends the jobs that support the iSeries Support for Windows Network Neighborhood server. This API does not end the QSERVER subsystem in which the jobs run.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority.

Required Parameter

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFB682 E	API &1 failed with reason code &2.
CPFB684 E	User does not have the correct authority for API &1.
CPFB689 E	Unable to end iSeries Support for Windows Network Neighborhood (iSeries NetServer).

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

End Server Session (QZLSENSS) API

Required Parameter Group:

1	Workstation name	Input	Char(15)
2	Error code	I/O	Char(*)

Optional Parameter:

3	Session identifier	Input	Binary(8)

Default Public Authority: *USE
Threadsafe: No

The End Server Session (QZLSENSS) API ends a single session or all the sessions that were established from the specified workstation.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority.

Required Parameter Group

Workstation name

INPUT; CHAR(15)

The name of the workstation from which the session to the server was established. When specifying the session identifier parameter to end a single session, *SESSID must be specified for the workstation name.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter

Session identifier

INPUT; BINARY(8)

Allows the caller to end a single session. This parameter must be 0 if any value other than *SESSID is specified for the workstation name. Allowable values are:

0	All sessions for the specified workstation name are ended.
Session identifier	The identified session is ended.

If the session identifier parameter is not specified, the API assumes a value of 0.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68E E	Error ending iSeries Support for Windows Network Neighborhood (iSeries NetServer) session.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

List Server Information (QZLSLSTI) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	Format	Input	Char(8)
3	Information qualifier	Input	Char(15)
4	Error code	I/O	Char(*)

Optional Parameter 1:

5	Session user	Input	Char(10)

Optional Parameter 2:

6	Session identifier	Input	Binary(8)

Default Public Authority: *USE
Threadsafe: No

The List Server Information (QZLSLSTI) API lists » share, configuration, session, statistics, and connection information for the server. « You may use the information qualifier function to list just the information in which you are interested.

Authorities and Locks

To use the *RESET qualifier for the ZLSL0400 format, you must have *IOSYSCFG special authority.

User Space Authority
*CHANGE

User Space Library Authority
» *EXECUTE «

User Space Lock
*EXCLRD

Required Parameter Group

Qualified user space name
INPUT; CHAR(20)

The user space that is to receive the server information. The first 10 characters contain the user space name, and the second 10 characters contain the name of the library where the user space is located.

Format

INPUT; CHAR(8)

The content and format of the information to be returned about the iSeries Support for Windows Network Neighborhood (iSeries NetServer) server. Possible format names follow:

ZLSL0100	Share information
ZLSL0101	Same as ZLSL0100, plus extension information
ZLSL0200	Configuration information
ZLSL0201	Same as ZLSL0200, plus additional fields
ZLSL0300	Session information
ZLSL0400	Statistical information
ZLSL0600	Session connection information
ZLSL0700	Share connection information
ZLSL0800	Share type information
ZLSL0900	Disabled user profiles

For more information, see "Format of Information List" on page 22.

Information qualifier

INPUT; CHAR(15)

A restriction on the information to be retrieved. This parameter allows the caller to request a subset of the entries that would normally be returned on a given format. Currently, this parameter applies only to the ZLSL0100 and ZLSL0101 (share information), ZLSL0300 (session information), ZLSL0400 (statistical information), ZLSL0600 (session connection information), ZLSL0700 (share connection information), and ZLSL0800 (share type information) formats.

For formats ZLSL0200 and ZLSL0201, this parameter is ignored.
For format ZLSL0300, a valid session name, *SESSID, or *ALL is allowed.
For format ZLSL0400, only the *ALL and *RESET qualifiers are valid.
For format ZLSL0600, a valid session name or *SESSID is required.
For format ZLSL0700, a valid share name is required.
For format ZLSL0800, only the *ALL qualifier is valid.
For format ZLSL0900 this parameter must be all blanks.

If this parameter is all blanks or is *ALL, no information qualification is performed. Allowable values include:

Information name	Information about the specified list entry only.
Generic information name	All information matching the generic string of the specified format.
**ALL*	All information of the specified format.
**RESET*	Statistical information is reset to zero. The information qualifier is valid for the ZLSL0400 format only. When the *RESET qualifier is specified, the values will be reset to zero before they are listed.
**SESSID*	The information returned is for the session indicated by the session identifier parameter. When this value is specified for this parameter, *SESSID must be specified for the session user parameter.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter 1

Session user

INPUT; CHAR(10)

Allows the caller to restrict the session information to be retrieved. This parameter is ignored if a generic information name, or *ALL, is specified on the information qualifier parameter. Currently, this parameter applies only to the ZLSL0300 (session information) and ZLSL0600 (session connection information) formats. Allowable values include:

**ALL*	The information returned is for all users of a specific session.
**SESSID*	The information returned is for the session indicated by the session identifier parameter. When this value is specified for this parameter, *SESSID must be specified for the information qualifier parameter.
Session user name	Information about the specified user only.

If the session user parameter is omitted, the API assumes a value of *ALL.

Optional Parameter 2

Session identifier

INPUT; BINARY(8)

Allows the caller to restrict the session information to be retrieved to a single session. This parameter must be 0 if any value other than *SESSID is specified on the information qualifier and session user parameters. Currently, this parameter applies only to the ZLSL0300 (session information) and ZLSL0600 (session connection information) formats. Allowable value are:

0	All information of the specified format
Session identifier	Information returned is for the specified session only

If the session identifier parameter is not specified, the API assumes a value of 0.

Format of Information List

The field list consists of:

- A user area
- A generic header section
- An input parameter section
- A data section
  - ZLSL0100
  - ZLSL0101
  - ZLSL0200
  - ZLSL0201
  - ZLSL0300
  - ZLSL0400
  - ZLSL0600
  - ZLSL0700
  - ZLSL0800
  - ZLSL0900

For information about the user area and generic header, see User Space Format for List APIs.

There are ten possible formats for the data section. The layout of the contents of the user space is determined by the format used. The following tables show how the contents of the input parameter section and the data format sections are organized. For descriptions of each field, see "Field Descriptions" on page 28.

When you retrieve list entry information from a user space, you must use the entry size returned in the generic header as a displacement to the next list entry for formats ZLSL0200, ZLSL0201, ZLSL0300, ZLSL0400, ZLSL0600, ZLSL0700, ZLSL0800, and ZLSL0900. For formats ZLSL0100 and ZLSL0101, use the length of this entry field at the start of each entry to calculate the displacement to the next list entry. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid.

Input Parameter Section

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	User space name specified
10	A	CHAR(10)	User space library name specified
20	14	CHAR(8)	Format specified
28	1C	CHAR(15)	Information qualifier specified

ZLSL0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of this entry
4	4	CHAR(12)	Share name
16	10	BINARY(4)	Device type
20	14	BINARY(4)	Permissions
24	18	BINARY(4)	Maximum users
28	1C	BINARY(4)	Current users
32	20	BINARY(4)	Spooled file type
36	24	BINARY(4)	Offset to path name
40	28	BINARY(4)	Length of path name
44	2C	CHAR(20)	Qualified output queue
64	40	CHAR(50)	Print driver type
114	72	CHAR(50)	Text description
164	A4	CHAR(*)	Path name

ZLSL0101 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of this entry

Offset		Type	Field
Dec	Hex		
4	4	CHAR(12)	Share name
16	10	BINARY(4)	Device type
20	14	BINARY(4)	Permissions
24	18	BINARY(4)	Maximum users
28	1C	BINARY(4)	Current users
32	20	BINARY(4)	Spooled file type
36	24	BINARY(4)	Offset to path name
40	28	BINARY(4)	Length of path name
44	2C	CHAR(20)	Qualified output queue
64	40	CHAR(50)	Print driver type
114	72	CHAR(50)	Text description
164	A4	CHAR(20)	Qualified printer file
184	B8	BINARY(4)	CCSID used for text conversion
188	BC	BINARY(4)	Offset to extension table
192	C0	BINARY(4)	Number of table entries
196	C4	CHAR(1)	Enable text conversion
197	C5	CHAR(1)	Publish print share
*	*	CHAR(*)	Path name
These fields repeat for each file extension table entry.		BINARY(4)	Length of file extension
		CHAR(46)	File extension

ZLSL0200 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	CCSID
4	4	BINARY(4)	Pending CCSID
8	8	BINARY(4)	Idle time-out
12	C	BINARY(4)	Pending idle time-out
16	10	BINARY(4)	Opportunistic lock time-out
20	14	BINARY(4)	Pending opportunistic lock time-out
24	18	BINARY(4)	Browsing interval
28	1C	BINARY(4)	Pending browsing interval
32	20	BINARY(4)	WINS enablement
36	24	BINARY(4)	Pending WINS enablement
40	28	BINARY(4)	Guest support
44	2C	BINARY(4)	Pending guest support
48	30	CHAR(10)	Guest user profile
58	3A	CHAR(10)	Pending guest user profile

Offset		Type	Field
Dec	Hex		
68	44	CHAR(15)	Server name
83	53	CHAR(15)	Pending server name
98	62	CHAR(15)	Domain name
113	71	CHAR(15)	Pending domain name
128	80	CHAR(50)	Text description
178	B2	CHAR(50)	Pending text description
228	E4	CHAR(15)	WINS primary address
243	F3	CHAR(15)	Pending WINS primary address
258	102	CHAR(15)	WINS secondary address
273	111	CHAR(15)	Pending WINS secondary address
288	120	CHAR(224)	Scope ID
512	200	CHAR(224)	Pending scope ID

ZLSL0201 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	CCSID
4	4	BINARY(4)	Pending CCSID
8	8	BINARY(4)	Idle time-out
12	C	BINARY(4)	Pending idle time-out
16	10	BINARY(4)	Opportunistic lock time-out
20	14	BINARY(4)	Pending opportunistic lock time-out
24	18	BINARY(4)	Browsing interval
28	1C	BINARY(4)	Pending browsing interval
32	20	BINARY(4)	WINS enablement
36	24	BINARY(4)	Pending WINS enablement
40	28	BINARY(4)	Guest support
44	2C	BINARY(4)	Pending guest support
48	30	CHAR(10)	Guest user profile
58	3A	CHAR(10)	Pending guest user profile
68	44	CHAR(15)	Server name
83	53	CHAR(15)	Pending server name
98	62	CHAR(15)	Domain name
113	71	CHAR(15)	Pending domain name
128	80	CHAR(50)	Text description
178	B2	CHAR(50)	Pending text description
228	E4	CHAR(15)	WINS primary address
243	F3	CHAR(15)	Pending WINS primary address

Offset		Type	Field
Dec	Hex		
258	102	CHAR(15)	WINS secondary address
273	111	CHAR(15)	Pending WINS secondary address
288	120	CHAR(224)	Scope ID
512	200	CHAR(224)	Pending scope ID
736	2E0	CHAR(1)	Allow system name
737	2E1	CHAR(1)	Pending allow system name
738	2E2	CHAR(1)	Authentication method
739	2E3	CHAR(1)	Pending authentication method
740	2E4	BINARY(4)	Server role
744	2E8	BINARY(4)	Pending server role
748	2EC	BINARY(4)	Message authentication
752	2F0	BINARY(4)	Pending message authentication
756	2F4	BINARY(4)	Minimum message severity
760	2F8	BINARY(4)	Pending minimum message severity
764	2FC	BINARY(4)	LAN Manager authentication
768	300	BINARY(4)	Pending LAN Manager authentication

ZLSL0300 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(15)	Workstation name
15	F	CHAR(10)	User profile name
25	19	CHAR(3)	Reserved
28	1C	BINARY(4)	Number of connections
32	20	BINARY(4)	Number of files open
36	24	BINARY(4)	Number of sessions
40	28	BINARY(4)	Session time
44	2C	BINARY(4)	Session idle time
48	30	CHAR(1)	Logon type
49	31	CHAR(1)	Encrypted password
50	32	CHAR(6)	Reserved
56	38	BINARY(8)	Session identifier

ZLSL0400 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	File opens

Offset		Type	Field
Dec	Hex		
4	4	BINARY(4)	Print jobs queued
8	8	BINARY(4)	Session starts
12	C	BINARY(4)	Sessions disconnected automatically
16	10	BINARY(4)	Sessions disconnected
20	14	CHAR(8)	Reserved
28	1C	BINARY(4)	Password violations
32	20	BINARY(4)	Guest support
36	24	BINARY(4)	Unknown users
40	28	BINARY(4)	Number of bytes sent (low)
44	2C	BINARY(4)	Number of bytes sent (high)
48	30	BINARY(4)	Number of bytes received (low)
52	34	BINARY(4)	Number of bytes received (high)
56	38	BINARY(4)	Average response time
60	3C	CHAR(13)	Date and time started
73	49	CHAR(13)	Date and time reset
86	56	CHAR(2)	Reserved

ZLSL0600 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Connection ID
4	4	BINARY(4)	Connection type
8	8	BINARY(4)	Number of files open
12	C	BINARY(4)	Number of connection users
16	10	BINARY(4)	Connection time
20	14	CHAR(10)	User name
30	1E	CHAR(12)	Share name
42	2A	CHAR(6)	Reserved
48	30	BINARY(8)	Session identifier
56	38	CHAR(8)	Reserved

ZLSL0700 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Connection ID
4	4	BINARY(4)	Connection type
8	8	BINARY(4)	Number of files open

Offset		Type	Field
Dec	Hex		
12	C	BINARY(4)	Number of connection users
16	10	BINARY(4)	Connection time
20	14	CHAR(10)	User name
30	1E	CHAR(15)	Workstation name
45	2D	CHAR(3)	Reserved
48	30	BINARY(8)	Session identifier
56	38	CHAR(8)	Reserved

ZLSL0800 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(12)	Share name
12	C	BINARY(4)	Device type

ZLSL0900 Format

Offset		Type	Field
Dec	Hex		
This field repeats for each user.		CHAR(10)	iSeries Netserver user name

Field Descriptions

Allow system name. Whether the server allows clients to connect using the system name found in the iSeries network attributes. Possible values are:

character 0 (0xF0) Do not allow access to the server using the system name.
character 1 (0xF1) Allow access to the server using the system name.

Authentication method. The method used to authenticate users. Possible values are:

character 0 (0xF0) The server authenticates with encrypted passwords only.
character 1 (0xF1) The server authenticates with network authentication only.
character 2 (0xF2) The server authenticates with network authentication when possible, but it allows clients to use encrypted passwords.

Average response time. The average server response time in milliseconds.

Browsing interval. The amount of time, in milliseconds, between each browsing server announcement.

CCSID. The coded character set ID (CCSID) that is used by all clients connected to the server.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

CCSID used for text conversion. The CCSID that is used for text file data conversion.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

Connection ID. The number that identifies the connection.

Connection time. The number of seconds since the connection was established.

Connection type. The type of the connection made from the workstation to the shared resource. Possible values are:

0 Disk drive
1 Spooled output queue

Current users. The number of connections that are currently made to the resource. A value of -1 (X'FFFFFFFF') means that the server was unable to retrieve the actual value.

Date and time reset. The date and time when the server statistics were reset.

The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 Date, in YYMMDD (year, month, and day) format.
8-13 Time of day, in HHMMSS (hours, minutes, and seconds) format.

Date and time started. The date and time when the server was started. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 Date, in YYMMDD (year, month, and day) format.
8-13 Time of day, in HHMMSS (hours, minutes, and seconds) format.

Device type. One of the following values:

0 Disk drive
1 Spooled output queue

Domain name. The name of the domain in which the iSeries Support for Windows Network Neighborhood server is a member.

Enable text conversion. Whether the server enables text conversion for this share. Possible values are:

0 Text conversion not enabled
1 Text conversion enabled
2 Text conversion enabled and mixed data is allowed

Encrypted password. Whether the the encrypted password was used to establish the session to the server. Possible values are:

0 An encrypted password was not used to establish the session
1 An encrypted password was used to establish the session

File extension. A file extension string that is null-terminated. Possible values include:

*	The server will convert all files.
.	The server will convert all files without an extension.
TXT, .TXT	The server will convert all files ending with .TXT (that is, a.TXT, a.b.c.TXT).
..TXT, ...TXT, ...TXT	Extensions with more than one leading period will have no effect on the server. No translation will be done.
*T*T*	The server will convert all files ending with an extension that substitutes any number of characters for the * wild card (that is, a.T123T, b.TXT, c.TEST).
T?T	The server will convert all files ending with an extension that substitutes any one character for the ? wild card (that is, a.T1T, b.TXT).

File opens. The number of file opens for the whole server.

Format specified. The content and format of the information to be returned about the iSeries Support for Windows Network Neighborhood server. Possible format names follow:

- ZLSL0100
- ZLSL0101
- ZLSL0200
- ZLSL0201
- ZLSL0300
- ZLSL0400
- ZLSL0600
- ZLSL0700
- ZLSL0800
- ZLSL0900

Guest support. A guest user profile may be used in the event an unknown user attempts to access resources on the system. Possible values are as follows:

0 Use of guest user profile not supported
1 Use of guest user profile supported

Guest user profile. The user profile to be used in the event an unknown user attempts to access resources on the system.

Idle time-out. The amount of time, in seconds, a connection to the simplified server remains active once activity has ceased on that connection.

Information qualifier specified. A restriction on the information to be retrieved. If a qualifier is specified that is inconsistent with the format requested, an error message is returned. A null list is returned if no information meets the qualifications. If this parameter is all blanks, no information qualification is performed. Allowable values include:

Information name	Information about this specified format only
Generic information name	All information that matches the generic string of the specified format
**ALL*	All information

**RESET* Statistical information is reset to zero. The information qualifier is valid for the ZLSL0400 format only.

iSeries NetServer user name. The name of an iSeries NetServer user.

LAN Manager (LANMAN) authentication. The level of restriction on the use of the LANMAN password hash for authentication. Possible values for the field follow:

0 LANMAN password hash is ignored if a stronger password hash is provided by the client
1 LANMAN password hash is used only if a stronger password hash provided by the client does not match or if a stronger password hash is not provided

Length of file extension. The size in bytes of the file extension. The length does not include the byte used for null-termination.

Length of path name. The length of the local path name of the shared resource in bytes.

Length of this entry. The total length of the list entry in bytes.

Logon type. The type of authentication used to connect to the server. Possible values for the flag follow:

0 Authentication using the defined guest user profile
1 Regular user authentication

Maximum users. The maximum number of concurrent connections that the shared resource can accommodate. This number is unlimited if the value is -1.

Message authentication. The status of message authentication. Possible values for the field follow:

0 Server does not support message authentication
1 Server supports message authentication. Message authentication is negotiated between the client and the server.
2 Server requires message authentication for all connections

Minimum message severity. The minimum message severity of administrative alerts to send to users of the server. Administrative alert messages are not sent if the value is -1.

Number of bytes received (high). The number of server bytes received from the network (high-order bits of 8-byte combinations of both number of bytes received fields).

Number of bytes received (low). The number of server bytes received from the network (low-order bits of 8-byte combinations of both number of bytes received fields).

Number of bytes sent (high). The number of server bytes sent to the network (high-order bits of 8-byte combinations of both number of bytes sent fields).

Number of bytes sent (low). The number of server bytes sent to the network (low-order bits of 8-byte combinations of both number of bytes sent fields).

Number of connections. The number of connections made during the session.

Number of connection users. The number of users on the connection.

Number of files open. The number of files that are open currently.

Number of sessions. The number of sessions that are established between the server and the requester.

Number of table entries. The number of file extension entries in the file extension table.

Offset to extension table. The offset from the start of user space where the file extension table starts.

Offset to path name. The offset from the start of user space where the path name of the shared resource starts.

» **Opportunistic lock (oplock) time-out.** The amount of time, in seconds, that the server will wait for a response to a break lock request sent to a lock holder before forcefully removing the lock. «

Password violations. The number of server password violations.

Path name. A path in the integrated file system. The path name is stored internally in unicode format; when listed, it is converted into the CCSID of the current job.

Pending allow system name. Whether the server allows clients to connect using the system name found in the network attributes the next time the server is started. Possible values are:

» *character 0 (0xF0)*	Do not allow access to the server using the system name
character 1 (0xF1)	Allow access to the server using the system name. «

Pending authentication method. The method that will be used to authenticate users the next time the server is started. Possible values are:

» *character 0 (0xF0)*	The server will authenticate with encrypted passwords only.
character 1 (0xF1)	The server will authenticate with network authentication only.
character 2 (0xF2)	The server will authenticate with network authentication when possible, but it will allow clients to use encrypted passwords when needed. «

Pending browsing interval. The amount of time, in seconds, to be used the next time the server is started, between each browsing server announcement.

Pending CCSID. The CCSID that is used by all clients connected to the server the next time the server is started.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

Pending domain name. The name of the domain in which the server will be a member the next time the server is started.

Pending guest support. A guest user profile may be used the next time the server is started in the event an unknown user attempts to access resources on the system. Possible values are:

0	Use of guest user profile not supported
1	Use of guest user profile supported

Pending guest user profile. The user profile to be used the next the time the server is started in the event that an unknown user attempts to access resources on the system.

Pending idle time-out. The amount of time (in seconds) to be used the next time the server is started so that a connection to the server will remain active once activity has ceased on that connection.

Pending LAN Manager (LANMAN) authentication. The level of restriction on the use of the LANMAN password hash to be used the next time the server is started. Possible values for the field follow:

0 LANMAN password hash will be ignored if a stronger password hash is provided by the client
1 LANMAN password hash will be used only if a stronger password hash provided by the client does not match or if a stronger password hash is not provided

Pending message authentication. The status of message authentication to be used the next time the server is started. Possible values are:

0 Server will not support message authentication
1 Server will support message authentication. Message authentication will be negotiated between the client and the server.
2 Server will require message authentication for all connections

Pending minimum message severity. The minimum message severity to be used the next time the server is started when sending administrative alerts to users of the server.

Pending opportunistic lock time-out. The amount of time, to be used the next time the server is started, that the server will wait for a response to a break lock request sent to a lock holder before forcefully removing the lock.

Pending scope ID. A string that names the network scope to be used by the Windows Internet Name Service (WINS) server the next time the server is started.

Pending server name. The name the server will be known as on the system the next time the server is started.

Pending server role. The value that determines the logon server role for this system the next time the server is started. Possible values are:

0 System will not be a server.
1 System will be a server.

Pending text description. An optional comment about the shared resource to be used the next time the server is started.

Pending WINS enablement. Whether this system uses the services of a Windows Internet Name Service (WINS) server the next time the server is started. Possible values are:

0 Do not use WINS server
1 Use WINS server

Pending WINS primary address. The IP address of the primary Windows Internet Name Service (WINS) server to be used the next time the server is started.

Pending WINS secondary address. The IP address of the secondary Windows Internet Name Service (WINS) server to be used the next time the server is started.

Permissions. Permissions to be applied against the disk for sharing. Possible values are:

1 Read only
2 Read/write

Print driver type. The text string that identifies the print driver appropriate for this share. When personal computers connect to this shared printer, this identifies the print driver they should use. This text should match the name of a print driver known to the personal computer operating system.

Print jobs queued. The number of server print jobs spooled.

Publish print share. Whether the print share is to be published. Possible values are:

0 Print share is not published
1 Print share is published

Qualified output queue. The name and library of the output queue. The first 10 characters identify the name of the output queue. The second 10 characters contain the library that contains the output queue.

Qualified printer file. The name and library of the printer file. The first 10 characters identify the name of the printer file. The second 10 characters contain the library that contains the printer file.

Reserved. Field not used currently.

Scope ID. A string that names the network scope to be used by the Windows Internet Name Service (WINS) server.

Server name. The name by which the server is known on the system.

Server role. The value that determines the logon server role for this system. Possible values are:

0 System is not a server.
1 System is a server.

Sessions disconnected. The number of server sessions disconnected normally or ended in error.

Sessions disconnected automatically. The number of server sessions disconnected automatically.

Session identifier. Unique identifier for the server session.

Session idle time. The number of seconds a session has been idle.

Session starts. The number of server session starts.

Session time. The number of seconds since the session was established.

Share name. The network name of the resource.

Spooled file type. The type of spooled files that will be created using this share. Values can be:

1 User ASCII (*USERASCII)

2 Advanced Function Printing (*AFP)
3 Simplified Character Set (*SCS)
4 Automatic type sensing

Text description. An optional comment about the shared resource or computer.

Unknown users. The number of unknown users that have requested sessions to the server.

User name. The name of the user that is associated with the connection.

User profile name. The name of the user that is associated with the resource.

User space library name specified. The user space library name that was passed to this API on the call in the qualified user space name parameter.

User space name specified. The user space name that was passed to this API on the call in the qualified user space name parameter.

WINS enablement. This system may use the services of a Windows Internet Name Service (WINS) server. Possible values are:

0 Do NOT use WINS server.
1 Use WINS server.

WINS primary address. The IP address of the primary Windows Internet Name Service (WINS) server.

WINS secondary address. The IP address of the secondary Windows Internet Name Service (WINS) server.

Workstation name. The name of the workstation from which the session to the server was established.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB686 E	Error configuring iSeries Support for Windows Network Neighborhood (iSeries NetServer).
CPFB68F E	Error listing iSeries Support for Windows Network Neighborhood (iSeries NetServer) information.
CPFB693 E	Data conversion failed for &5 API.
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Open List of Server Information (QZLSOLST) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(64)
4	Format	Input	Char(8)
5	Information qualifier	Input	Char(15)
6	Error code	I/O	Char(*)

Optional Parameter 1:

7	Session user	Input	Char(10)

Optional Parameter 2:

8	Session identifier	Input	Binary(8)

Default Public Authority: *USE
Threadsafe: No

The Open List of Server Information (QZLSOLST) API opens a list of information about the server for share, configuration, session, statistics, or connection information. You may use the information qualifier function to open a list of just the information in which you are interested.

Authorities and Locks

To use the *RESET qualifier for the ZLSL0400 format, you must have *IOSYSCFG special authority.

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The variable used to return the server information. For details about the structure of the receiver variable, see "Format of Receiver Variable" on page 39.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable.

List information
OUTPUT; CHAR(64)

The information about the list of information that was opened. For a layout of this structure, see "Format of List Information" on page 38.

Format
INPUT; CHAR(8)

The content and format of the information to be returned about the iSeries Support for Windows Network Neighborhood » (iSeries NetServer) « server. Possible format names follow:

ZLSL0100	Share information
ZLSL0101	Same as ZLSL0100, plus extension information
ZLSL0200	Configuration information
ZLSL0201	Same as ZLSL0200, plus » additional fields «
ZLSL0300	Session information
ZLSL0400	Statistical information
ZLSL0600	Session connection information

ZLSL0700	Share connection information
ZLSL0800	Share type information
ZLSL0900	Disabled user profiles

For more information, see "Format of Receiver Variable" on page 39.

Information qualifier

INPUT; CHAR(15)

A restriction on the information to be retrieved. This parameter allows the caller to request a subset of the entries that would normally be returned on a given format. Currently, this parameter applies only to the ZLSL0100 and ZLSL0101 (share information), ZLSL0300 (session information), ZLSL0400 (statistical information), ZLSL0600 (session connection information), ZLSL0700 (share connection information), and ZLSL0800 (share type information) formats.

For formats ZLSL0200 and ZLSL0201, this parameter is ignored.
For format ZLSL0300, a valid session name, *SESSID, or *ALL is allowed.
For format ZLSL0400, only the *ALL and *RESET qualifiers are valid.
For format ZLSL0600, a valid session name or *SESSID is required.
For format ZLSL0700, a valid share name is required.
For format ZLSL0800, only the *ALL qualifier is valid.
For format ZLSL0900 this parameter must be all blanks.

If this parameter is all blanks or is *ALL, no information qualification is performed. Allowable values include:

Information name	Information about the specified list entry only.
Generic information name	All information matching the generic string of the specified format.
**ALL*	All information of the specified format.
**RESET*	Statistical information is reset to zero. This parameter is valid for the ZLSL0400 format only. When the *RESET qualifier is specified, the values will be reset to zero before they are listed.
**SESSID*	The information returned is for the session indicated by the session identifier parameter. When this value is specified for this parameter, *SESSID must be specified for the session user parameter.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Optional Parameter 1

Session user

INPUT; CHAR(10)

Allows the caller to restrict the session information to be retrieved. This parameter is ignored if a generic information name, or *ALL, is specified on the information qualifier parameter. Currently, this parameter applies only to the ZLSL0300 (session information) and ZLSL0600 (session connection information) formats. Allowable values include:

**ALL*	The information returned is for all users of a specific session.
**SESSID*	The information returned is for the session indicated by the session identifier parameter. When this value is specified for this parameter, *SESSID must be specified for the information qualifier parameter.
Session user name	Information about the specified user only.

If the session user parameter is omitted, the API assumes a value of *ALL.

Optional Parameter 2

Session identifier
INPUT; BINARY(8)

Allows the caller to restrict the session information to be retrieved to a single session. This parameter must be 0 if any value other than *SESSID is specified on the information qualifier and session user parameters. Currently, this parameter applies only to the ZLSL0300 (session information) and ZLSL0600 (session connection information) formats. Allowable value are:

0 All information of the specified format.
Session identifier Information returned is for the specified session only.

If the session identifier parameter is not specified, the API assumes a value of 0.

Format of List Information

The following table shows the format of the list information parameter. For detailed descriptions of the fields in the table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	BINARY(4)	Record length
12	C	BINARY(4)	Length of information returned
16	10	CHAR(1)	Information complete indicator
17	11	CHAR(13)	Date and time created
30	1E	CHAR(34)	Reserved

Field Descriptions

Date and time created. The date and time when the list was created. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 Date, in YYMMDD (year, month, and day) format.
8-13 Time of day, in HHMMSS (hours, minutes, and seconds) format.

Information complete indicator. Whether all information requested has been supplied.

C Complete and accurate information. All the buffers requested are read and returned.
I Incomplete information. An interruption causes the list to contain incomplete information about a buffer or buffers.

Length of information returned. The size in bytes of the information returned in the receiver variable.

Record length. The length of each record of information returned. For variable length records, this value will be set to zero. For variable length records, you can obtain the length of individual records from the records themselves.

Records returned. The number of records returned in the receiver variable.

Reserved. Field not used currently. It should be initialized to hexadecimal zeroes (x'00's).

Total records. The total number of records available in the list.

Format of Receiver Variable

There are ten possible formats for the receiver variable:

- ZLSL0100
- ZLSL0101
- ZLSL0200
- ZLSL0201
- ZLSL0300
- ZLSL0400
- ZLSL0600
- ZLSL0700
- ZLSL0800
- ZLSL0900

The layout of the contents of the receiver variable is determined by the format used. The following tables show how the contents of the receiver variable is organized. For descriptions of each field, see "Field Descriptions" on page 44.

When you retrieve list entry information from the receiver variable, you must use the entry size returned in the list information as a displacement to the next list entry for formats ZLSL0200, ZLSL0201, ZLSL0300, ZLSL0400, ZLSL0600, ZLSL0700, ZLSL0800, and ZLSL0900. For formats ZLSL0100 and ZLSL0101, use the length of this entry field at the start of each entry to calculate the displacement to the next list entry. The size of each entry may be padded at the end. If you do not use the entry size, the result may not be valid.

ZLSL0100 Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of this entry
4	4	CHAR(12)	Share name
16	10	BINARY(4)	Device type
20	14	BINARY(4)	Permissions
24	18	BINARY(4)	Maximum users
28	1C	BINARY(4)	Current users
32	20	BINARY(4)	Spooled file type
36	24	BINARY(4)	Offset to path name (ZLSL0100 format)
40	28	BINARY(4)	Length of path name
44	2C	CHAR(20)	Qualified output queue
64	40	CHAR(50)	Print driver type
114	72	CHAR(50)	Text description
164	A4	CHAR(*)	Path name

ZLSL0101 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of this entry
4	4	CHAR(12)	Share name
16	10	BINARY(4)	Device type
20	14	BINARY(4)	Permissions
24	18	BINARY(4)	Maximum users
28	1C	BINARY(4)	Current users
32	20	BINARY(4)	Spooled file type
36	24	BINARY(4)	Offset to path name (ZLSL0101 format)
40	28	BINARY(4)	Length of path name
44	2C	CHAR(20)	Qualified output queue
64	40	CHAR(50)	Print driver type
114	72	CHAR(50)	Text description
164	A4	CHAR(20)	Qualified printer file
184	B8	BINARY(4)	CCSID used for text conversion
188	BC	BINARY(4)	Offset to extension table
192	C0	BINARY(4)	Number of table entries
196	C4	CHAR(1)	Enable text conversion
197	C5	CHAR(1)	Publish print share
*	*	CHAR(*)	Path name
These fields repeat for each file extension table entry.		BINARY(4)	Length of file extension
		CHAR(46)	File extension

ZLSL0200 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	CCSID
4	4	BINARY(4)	Pending CCSID
8	8	BINARY(4)	Idle time-out
12	C	BINARY(4)	Pending idle time-out
16	10	BINARY(4)	Opportunistic lock time-out
20	14	BINARY(4)	Pending opportunistic lock time-out
24	18	BINARY(4)	Browsing interval
28	1C	BINARY(4)	Pending browsing interval
32	20	BINARY(4)	WINS enablement
36	24	BINARY(4)	Pending WINS enablement
40	28	BINARY(4)	Guest support

Offset		Type	Field
Dec	Hex		
44	2C	BINARY(4)	Pending guest support
48	30	CHAR(10)	Guest user profile
58	3A	CHAR(10)	Pending guest user profile
68	44	CHAR(15)	Server name
83	53	CHAR(15)	Pending server name
98	62	CHAR(15)	Domain name
113	71	CHAR(15)	Pending domain name
128	80	CHAR(50)	Text description
178	B2	CHAR(50)	Pending text description
228	E4	CHAR(15)	WINS primary address
243	F3	CHAR(15)	Pending WINS primary address
258	102	CHAR(15)	WINS secondary address
273	111	CHAR(15)	Pending WINS secondary address
288	120	CHAR(224)	Scope ID
512	200	CHAR(224)	Pending scope ID

ZLSL0201 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	CCSID
4	4	BINARY(4)	Pending CCSID
8	8	BINARY(4)	Idle time-out
12	C	BINARY(4)	Pending idle time-out
16	10	BINARY(4)	Opportunistic lock time-out
20	14	BINARY(4)	Pending opportunistic lock time-out
24	18	BINARY(4)	Browsing interval
28	1C	BINARY(4)	Pending browsing interval
32	20	BINARY(4)	WINS enablement
36	24	BINARY(4)	Pending WINS enablement
40	28	BINARY(4)	Guest support
44	2C	BINARY(4)	Pending guest support
48	30	CHAR(10)	Guest user profile
58	3A	CHAR(10)	Pending guest user profile
68	44	CHAR(15)	Server name
83	53	CHAR(15)	Pending server name
98	62	CHAR(15)	Domain name
113	71	CHAR(15)	Pending domain name
128	80	CHAR(50)	Text description

Offset		Type	Field
Dec	Hex		
178	B2	CHAR(50)	Pending text description
228	E4	CHAR(15)	WINS primary address
243	F3	CHAR(15)	Pending WINS primary address
258	102	CHAR(15)	WINS secondary address
273	111	CHAR(15)	Pending WINS secondary address
288	120	CHAR(224)	Scope ID
512	200	CHAR(224)	Pending scope ID
736	2E0	CHAR(1)	Allow system name
737	2E1	CHAR(1)	Pending allow system name
738	2E2	CHAR(1)	Authentication method
739	2E3	CHAR(1)	Pending authentication method
740	2E4	BINARY(4)	Server role
744	2E8	BINARY(4)	Pending server role
748	2EC	BINARY(4)	Message authentication
752	2F0	BINARY(4)	Pending message authentication
756	2F4	BINARY(4)	Minimum message severity
760	2F8	BINARY(4)	Pending minimum message severity
764	2FC	BINARY(4)	LAN Manager authentication
768	300	BINARY(4)	Pending LAN Manager authentication

ZLSL0300 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(15)	Workstation name
15	F	CHAR(10)	User profile name
25	19	CHAR(3)	Reserved
28	1C	BINARY(4)	Number of connections
32	20	BINARY(4)	Number of files open
36	24	BINARY(4)	Number of sessions
40	28	BINARY(4)	Session time
44	2C	BINARY(4)	Session idle time
48	30	CHAR(1)	Logon type
49	31	CHAR(1)	Encrypted password
50	32	CHAR(6)	Reserved
56	38	BINARY(8)	Session identifier

ZLSL0400 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	File opens
4	4	BINARY(4)	Print jobs queued
8	8	BINARY(4)	Session starts
12	C	BINARY(4)	Sessions disconnected automatically
16	10	BINARY(4)	Sessions disconnected
20	14	CHAR(8)	Reserved
28	1C	BINARY(4)	Password violations
32	20	BINARY(4)	Guest support
36	24	BINARY(4)	Unknown users
40	28	BINARY(4)	Number of bytes sent (low)
44	2C	BINARY(4)	Number of bytes sent (high)
48	30	BINARY(4)	Number of bytes received (low)
52	34	BINARY(4)	Number of bytes received (high)
56	38	BINARY(4)	Average response time
60	3C	CHAR(13)	Date and time started
73	49	CHAR(13)	Date and time reset
86	56	CHAR(2)	Reserved

ZLSL0600 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Connection ID
4	4	BINARY(4)	Connection type
8	8	BINARY(4)	Number of files open
12	C	BINARY(4)	Number of connection users
16	10	BINARY(4)	Connection time
20	14	CHAR(10)	User name
30	1E	CHAR(12)	Share name
42	2A	CHAR(6)	Reserved
48	30	BINARY(8)	Session identifier
56	38	CHAR(8)	Reserved

ZLSL0700 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Connection ID
4	4	BINARY(4)	Connection type
8	8	BINARY(4)	Number of files open
12	C	BINARY(4)	Number of connection users
16	10	BINARY(4)	Connection time
20	14	CHAR(10)	User name
30	1E	CHAR(15)	Workstation name
45	2D	CHAR(3)	Reserved
48	30	BINARY(8)	Session identifier
56	38	CHAR(8)	Reserved

ZLSL0800 Format

Offset		Type	Field
Dec	Hex		
0	0	CHAR(12)	Share name
12	C	BINARY(4)	Device type

ZLSL0900 Format

Offset		Type	Field
Dec	Hex		
This field repeats for each user.		CHAR(10)	iSeries Netserver user name

Field Descriptions

Allow system name. Whether the server allows clients to connect using the system name found in the iSeries network attributes. Possible values are:

character 0 (0xF0) Do not allow access to the server using the system name.
character 1 (0xF1) Allow access to the server using the system name.

Authentication method. The method used to authenticate users. Possible values are:

character 0 (0xF0) The server authenticates with encrypted passwords only.
character 1 (0xF1) The server authenticates with network authentication only.
character 2 (0xF2) The server authenticates with network authentication when possible, but it allows clients to use encrypted passwords.

Average response time. The average server response time in milliseconds.

Browsing interval. The amount of time, in milliseconds, between each browsing server announcement.

CCSID. The coded character set ID (CCSID) that is used by all clients connected to the server.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

CCSID used for text conversion. The CCSID that is used for text file data conversion.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

Connection ID. The number that identifies the connection.

Connection time. The number of seconds since the connection was established.

Connection type. The type of the connection made from the workstation to the shared resource. Possible values are:

0 Disk drive
1 Spooled output queue

Current users. The number of connections that are currently made to the resource. A value of -1 (X'FFFFFFFF') means that the server was unable to retrieve the actual value.

Date and time reset. The date and time when the server statistics were reset. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 Date, in YYMMDD (year, month, and day) format.
8-13 Time of day, in HHMMSS (hours, minutes, and seconds) format.

Date and time started. The date and time when the server was started. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 Date, in YYMMDD (year, month, and day) format.
8-13 Time of day, in HHMMSS (hours, minutes, and seconds) format.

Device type. One of the following values:

0 Disk drive
1 Spooled output queue

Domain name. The name of the domain in which the iSeries Support for Windows Network Neighborhood server is a member.

Enable text conversion. Whether the server enables text conversion for this share. Possible values are:

0 Text conversion not enabled
1 Text conversion enabled
2 Text conversion enabled and mixed data is allowed

Encrypted password. Whether the the encrypted password was used to establish the session to the server. Possible values are:

0 An encrypted password was not used to establish the session
1 An encrypted password was used to establish the session

File extension. A file extension string that is null-terminated. Possible values include:

*	The server will convert all files.
.	The server will convert all files without an extension.
TXT, .TXT	The server will convert all files ending with .TXT (that is, a.TXT, a.b.c.TXT).
..TXT, ...TXT, ...TXT	Extensions with more than one leading period will have no effect on the server. No translation will be done.
*T*T*	The server will convert all files ending with an extension that substitutes any number of characters for the * wild card (that is, a.T123T, b.TXT, c.TEST).
T?T	The server will convert all files ending with an extension that substitutes any one character for the ? wild card (that is, a.T1T, b.TXT).

File opens. The number of file opens for the whole server.

Guest support. A guest user profile may be used in the event an unknown user attempts to access resources on the system. Possible values are:

0 Use of guest user profile not supported
1 Use of guest user profile supported

Guest user profile. The user profile to be used in the event an unknown user attempts to access resources on the system.

Idle time-out. The amount of time, in seconds, a connection to the simplified server remains active once activity has ceased on that connection.

iSeries NetServer user name. The name of an iSeries NetServer user.

» **LAN Manager (LANMAN) authentication.** The level of restriction on the use of the LANMAN password hash for authentication. Possible values for the field follow:

0 LANMAN password hash is ignored if a stronger password hash is provided by the client
1 LANMAN password hash is used only if a stronger password hash provided by the client does not match or if a stronger password hash is not provided «

Length of file extension. The size in bytes of the file extension. The length does not include the byte used for null-termination.

Length of path name. The length of the local path name of the shared resource in bytes.

Length of this entry. The total length of the list entry in bytes.

Logon type. The type of authentication used to connect to the server. Possible values are:

0 Authentication using the defined guest user profile
1 Regular user authentication

Maximum users. The maximum number of concurrent connections that the shared resource can accommodate. This number is unlimited if the value is -1.

» **Message authentication.** The status of message authentication. Possible values for the field follow:

0 Server does not support message authentication
1 Server supports message authentication. Message authentication is negotiated between the client and the server.
2 Server requires message authentication for all connections

Minimum message severity. The minimum message severity of administrative alerts to send to users of the server. Administrative alert messages are not sent if the value is -1. «

Number of bytes received (high). The number of server bytes received from the network (high-order bits of 8-byte combinations of both number of bytes received fields).

Number of bytes received (low). The number of server bytes received from the network (low-order bits of 8-byte combinations of both number of bytes received fields).

Number of bytes sent (high). The number of server bytes sent to the network (high-order bits of 8-byte combinations of both number of bytes sent fields).

Number of bytes sent (low). The number of server bytes sent to the network (low-order bits of 8-byte combinations of both number of bytes sent fields).

Number of connections. The number of connections made during the session.

Number of connection users. The number of users on the connection.

Number of files open. The number of files that are open currently.

Number of sessions. The number of sessions that are established between the server and the requester.

Number of table entries. The number of file extension entries in the file extension table.

Offset to extension table. The offset from the start of returned data where the file extension table starts.

Offset to path name (ZLSL0100 format). The offset from the start of the record to where the path name of the shared resource starts.

Offset to path name (ZLSL0101 format). The offset from the start of the returned data to where the path name of the shared resource starts.

» **Opportunistic lock (oplock) time-out.** The amount of time, in seconds, that the server will wait for a response to a break lock request sent to a lock holder before forcefully removing the lock. «

Password violations. The number of server password violations.

Path name. A path in the integrated file system. The path name is stored internally in unicode format; when listed, it is converted into the CCSID of the current job.

Pending allow system name. Whether the server allows clients to connect using the system name found in the network attributes the next time the server is started. Possible values are:

character 0 (0xF0) Do not allow access to the server using the system name

character 1 (0xF1) Allow access to the server using the system name.

Pending authentication method. The method that will be used to authenticate users the next time the server is started. Possible values are:

character 0 (0xF0) The server will authenticate with encrypted passwords only.

character 1 (0xF1) The server will authenticate with network authentication only.

character 2 (0xF2) The server will authenticate with network authentication when possible, but it will allow clients to use encrypted passwords when needed.

Pending browsing interval. The amount of time, in seconds, to be used the next time the server is started, between each browsing server announcement.

Pending CCSID. The CCSID that is used by all clients connected to the server the next time the server is started.

Note: The field in the corresponding header file for this format may indicate code page, but the value returned will be the CCSID.

Pending domain name. The name of the domain in which the server will be a member the next time the server is started.

Pending guest support. A guest user profile may be used the next time the server is started in the event an unknown user attempts to access resources on the system. Possible values are:

0 Use of guest user profile not supported.

1 Use of guest user profile supported.

Pending guest user profile. The user profile to be used the next the time the server is started in the event that an unknown user attempts to access resources on the system.

Pending idle time-out. The amount of time (in seconds) to be used the next time the server is started so that a connection to the server will remain active once activity has ceased on that connection.

Pending LAN Manager (LANMAN) authentication. The level of restriction on the use of the LANMAN password hash to be used the next time the server is started. Possible values for the field follow:

0 LANMAN password hash will be ignored if a stronger password hash is provided by the client

1 LANMAN password hash will be used only if a stronger password hash provided by the client does not match or if a stronger password hash is not provided

Pending message authentication. The status of message authentication to be used the next time the server is started. Possible values are:

0 Server will not support message authentication

1 Server will support message authentication. Message authentication will be negotiated between the client and the server.

2 Server will require message authentication for all connections

Pending minimum message severity. The minimum message severity to be used the next time the server is started when sending administrative alerts to users of the server.

Pending opportunistic lock time-out. The amount of time, to be used the next time the server is started, that the server will wait for a response to a break lock request sent to a lock holder before forcefully removing the lock. «

Pending scope ID. A string that names the network scope to be used by the Windows Internet Name Service (WINS) server the next time the server is started.

Pending server name. The name the server will be known as on the system the next time the server is started.

Pending server role. The value that determines the logon server role for this system the next time the server is started. Possible values are:

0 System will not be a server.
1 System will be a server.

Pending text description. An optional comment about the shared resource to be used the next time the server is started.

Pending WINS enablement. Whether this system uses the services of a Windows Internet Name Service (WINS) server the next time the server is started. Possible values are:

0 Do not use WINS server
1 Use WINS server

Pending WINS primary address. The IP address of the primary Windows Internet Name Service (WINS) server to be used the next time the server is started.

Pending WINS secondary address. The IP address of the secondary Windows Internet Name Service (WINS) server to be used the next time the server is started.

Permissions. Permissions to be applied against the disk for sharing. Possible values are:

1 Read only
2 Read/write

Print driver type. The text string that identifies the print driver appropriate for this share. When personal computers connect to this shared printer, this identifies the print driver they should use. This text should match the name of a print driver known to the personal computer operating system.

Print jobs queued. The number of server print jobs spooled.

Publish print share. Whether the print share is to be published. Possible values are:

0 Print share is not published
1 Print share is published

Qualified output queue. The name and library of the output queue. The first 10 characters identify the name of the output queue. The second 10 characters contain the library that contains the output queue.

Qualified printer file. The name and library of the printer file. The first 10 characters identify the name of the printer file. The second 10 characters contain the library that contains the printer file.

Reserved. Field not used currently.

Scope ID. A string that names the network scope to be used by the Windows Internet Name Service (WINS) server.

Server name. The name by which the server is known on the system.

Server role. The value that determines the logon server role for this system. Possible values are:

0 System is not a server.
1 System is a server.

Sessions disconnected. The number of server sessions disconnected normally or ended in error.

Sessions disconnected automatically. The number of server sessions disconnected automatically.

Session identifier. Unique identifier for the server session.

Session idle time. The number of seconds a session has been idle.

Session starts. The number of server session starts.

Session time. The number of seconds that a session has been active.

Share name. The network name of the resource.

Spooled file type. The type of spooled files that will be created using this share. Values can be:

1 User ASCII (*USERASCII)
2 Advanced Function Printing (*AFP)
3 Simplified Character Set (*SCS)
4 Automatic type sensing

Text description. An optional comment about the shared resource or computer.

Unknown users. The number of unknown users that have requested sessions to the server.

User name. The name of the user that is associated with the connection.

User profile name. The name of the user that is associated with the resource.

WINS enablement. This system may use the services of a Windows Internet Name Service (WINS) server. Possible values are:

0 Do NOT use WINS server.
1 Use WINS server.

WINS primary address. The IP address of the primary Windows Internet Name Service (WINS) server.

WINS secondary address. The IP address of the secondary Windows Internet Name Service (WINS) server.

Workstation name. The name of the workstation from which the session to the server was established.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPFB688 E	Error opening iSeries Support for Windows Network Neighborhood (iSeries NetServer) information.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Remove Server Share (QZLSRMS) API

Required Parameter Group:

1	Share name	Input	Char(12)
2	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Remove Server Share (QZLSRMS) API removes a share from the list of files or printers on the server. When this function removes a share, all connections to the shared resource are disconnected.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority, own the integrated file system directory, or own the system output queue that the share references.

Required Parameter Group

Share name
INPUT; CHAR(12)

The name of the share to be removed.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFA0D4 E	File system error occurred.
CPFB682 E	API &1 failed with reason code &2.
CPFB683 E	Data conversion failed for API &1.
CPFB684 E	User does not have the correct authority for API &1.
CPFB68A E	Error occurred while working with shared resource &2.
CPFB68B E	Character is not valid for value &3.
CPFB693 E	Data conversion failed for &5 API.

Message ID	Error Message Text
CPIB685 E	Error occurred on iSeries Support for Windows Network Neighborhood (iSeries NetServer) request.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Start Server (QZLSSTRS) API

Required Parameter Group:

1	Reset qualifier	Input	Char(1)
2	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Start Server (QZLSSTRS) API starts the jobs necessary for the iSeries Support for Windows Network Neighborhood server to run. If the jobs are already running, the API will fail.

This API does not start the QSERVER subsystem (in which the jobs run). If the QSERVER subsystem is not running, the API will fail.

Authorities and Locks

To use this API, you must have *IOSYSCFG special authority.

Required Parameter Group

Reset qualifier

INPUT; CHAR(1)

Whether the server is reset when started. Possible values are:

0 Server not reset
1 Server reset

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Error Messages

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPFB682 E	API &1 failed with reason code &2.
CPFB684 E	User does not have the correct authority for API &1.
CPFB685 E	iSeries Support for Windows Network Neighborhood (iSeries NetServer) was unable to start.

API introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

Integrated Operating Environments APIs

The Integrated Operating Environments APIs provide information about operating environments, such as Windows(R) or Linux, that are integrated with i5/OS(TM). For example, attributes of integrated servers can be retrieved, as well as attributes of virtual disk drives and users that are associated with integrated servers.

In addition, to the integrated Windows(R) or Linux APIs, there are some APIs which allow you to manage monitored resources. A resource may be a system object or a set of attributes not associated with a specific system object, such as the set of system environment variables. System objects and attributes are monitored by the system to allow notifications of changes to the attributes to be passed to a supporting system function or application for system management purposes.

The Integrated Operating Environments APIs are:

- "Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API" (QfpadAddMonitoredResourceEntry) The Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API adds a monitored resource entry for a system resource and its attributes.
- "Open List Network Server Storage Spaces (QFPADOLD) API" on page 60 (QFPADOLD) retrieves a list of storage spaces (virtual disk drives) that can be linked to a server.
- "Open List Network Server Users (QFPADOLU) API" on page 68 (QFPADOLU) generates a list of Windows domains and servers, and the associated profiles enrolled to them in the network.
- "Open List Network Servers (QFPADOLS) API" on page 79 (QFPADOLS) retrieves a list of servers.
- "Process Network Server User (QFPADPNU) API" on page 88 (QFPADPNU) performs an error recovery operation on a selected user or group enrollment entry for a particular Windows domain or server.
- "Remove Monitored Resource Entry (QfpadRmvMonitoredResourceEntry) API" on page 90 (QfpadRmvMonitoredResourceEntry) The Remove Monitored Resource Entry (QfpadRmvMonitoredResourceEntry) API removes an Monitored Resource Entry (MRE) from the monitored resource directory.
- "Retrieve Monitored Resource Information (QfpadRtvMonitoredResourceInfo) API" on page 95 (QfpadRtvMonitoredResourceInfo) The Retrieve Monitored Resource Information (QfpadRtvMonitoredResourceInfo) API returns information about monitored resources.
- "Retrieve Network Server Information (QFPADRNI) API" on page 105 (QFPADRNI) retrieves information from a Windows server.
- "Retrieve Network Server User Attributes (QFPADRUA) API" on page 108 (QFPADRUA) retrieves the network server attributes for a user or a group.
- Retrieve Remote Network Server Status (QITDRSTS) retreives the power and operating system state of a remote server.

Top | "Server Support APIs," on page 1 | APIs by category

Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API

Required Parameter Group:

1	Manager type	Input	Char(10)
2	Monitored resource information	Input	Char(*)
3	Length of monitored resource information	Input	Binary(4)
4	Format of monitored resource information	Input	Char(8)
5	Attribute information	Input	Char(*)

6	Length of attribute information	Input	Binary(4)
7	Format of attribute information	Input	Char(8)
8	Server information	Input	Char(*)
9	Length of server information	Input	Binary(4)
10	Format of server information	Input	Char(8)
11	Server defined output	Output	Char(*)
12	Error code	I/O	Char(*)

Service Program Name: QFPADAP1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API adds a monitored resource entry for a system resource and its attributes. A resource may be a system object or a set of attributes not associated with a specific system object, such as the set of system environment variables. System objects and attributes are monitored by the system to allow notifications of changes to the attributes to be passed to a supporting system function or application for system management purposes.

When adding a monitored resource entry (MRE) for a system object, the monitored resource name is the name of the system object. It is possible to specify one or more of the system object's attributes for monitoring. Otherwise, a special value of -1 may be specified as the number of attribute entries, which will cause all of the system object's allowed attributes to be monitored by the system.

When adding a monitored resource entry (MRE) for a resource that is not a system object, such as a system value, network attribute, TCP/IP attribute or system environment variable, the monitored resource name is the name of the specific system value, network attribute, TCP/IP attribute, or system environment variable which is to be monitored by the system.

This API will do the following when the manager type is *ADMDMN:

- Create a monitored resource entry on all nodes in the cluster administrative domain.
- The resource being monitored by the cluster administrative domain will be created on all nodes in the cluster administrative domain where it does not already exist. The created attributes will be set to the values of the attributes from the monitored resource on the node from which the API was called.
- If the cluster administrative domain has been started, the values of the attributes for the monitored resource will be synchronized to the same attribute values from the monitored resource on the node where the API was called.
- If the cluster administrative domain has not been started, the attribute values for the resource being monitored by the cluster administrative domain will not be synchronized until the cluster administrative domain is started.
- Once the API has completed, if the cluster administrative domain is not active, the global status will be ADDED. If the cluster administative domain is active, the global status will be either PENDING, CONSISTENT or INCONSISTENT.

This API requires the following when the manager type is *ADMDMN:

- When adding a monitored resource entry (MRE) to a cluster administrative domain, the cluster administrative domain must exist and all nodes in the domain must be active.
- Resources cannot be added to a cluster administrative domain when it is partitioned.
- To add a monitored resource entry (MRE), the monitored resource must exist on the system from which this API is called.
- The program that calls this API must be running under a user profile that exists on all nodes in the cluster administrative domain.
- This API operates in an asynchronous mode. See Behavior of Cluster Resource Services APIs for more information.

Authorities and Locks

The following authorities are explicitly checked for by this API. Additional authority checking may be performed by APIs or CL commands called as a result of this API. See the Monitored resources topic in the information center for more information.

General Authority

The program that calls this API must be running under a user profile with system configuration (*IOSYSCFG) special authority.

The caller of the API will need authority to retrieve the value of the monitored resource entry's attributes on the source system and authority to create or change the monitored resource entry.

- Security Administrator (*SECADM) special authority is required to add a monitored resource entry (MRE) for user profile.
- Job Control (*JOBCTL) special authority is required to add a monitored resource entry (MRE) for system environment variable.
- Authority required to change the specified system value, is required to add a monitored resource entry (MRE) for system value. *ALLOBJ authority is required for system values QACGLVL, QDEVNAMING, QDSPSGNINF, QKBDBUF, QLMTSECOFR, QMAXJOB, QMAXSGNACN, QMAXSIGN, QPWDEXPITV, QPWDLMTAJC, QPWDLMTCHR, QPWDLMTREP, QPWDLVL, QPWDMAXLEN, QPWDMINLEN, QPWDPOSDIF, QPWDRQDDGT, QPWDRQDDIF, QRETSVRSEC, QSECURITY, and QSPLFACN.
- *ALLOBJ authority is required for network attributes DDMACC or PSCACC.
- User Profile Authority (applies to the user profile for which the MRE is added) *USE and *OBJMGT.
- IASP Device Description Authority (applies to a configuration object for which the MRE is added) *CHANGE, *ADD, *READ and *OBJMGT.
- Library Authority (applies to the library for any object for which the MRE is added) *EXECUTE.
- Job Description Authority (applies to a job description for which the MRE is added) *OBJMGT and *OBJOPR.
- Class Authority (applies to a class object for which the MRE is added) *OBJMGT and *OBJOPR.

User queue

The following authorities and locks are required for the user queue specified in the results information parameter.

User Queue Authority

*OBJOPR and *ADD

User Queue Library Authority

*EXECUTE

User Queue Lock

*EXCLRD

Required Parameter Group

Manager type

INPUT; CHAR(10)

The management function which is to monitor the resource. *ADMDMN must be specified.

Monitored resource information

INPUT; CHAR(*)

The monitored resource information provided as input.

Length of monitored resource information
INPUT; BINARY(4)

The length of the monitored resource information.

Format of monitored resource information
INPUT; CHAR(8)

The format name of the monitored resource information. The following format name is allowed:

"Monitored Resource Entry (EENT0100 Format)" on page 57 The format of the monitored resource information.

Attribute information
INPUT; CHAR(*)

The attribute information for the resource being monitored.

Length of attribute information
INPUT; BINARY(4)

The length of attribute information.

Format of attribute information
INPUT; CHAR(8)

The format name for the attribute information. The following format name is allowed:

"Attribute Information (ATRI0100 Format)" on page 57 The format of the attribute information for an entry.

Server information
INPUT; CHAR(*)

The information about the server where the monitored resource entry will be monitored by the system. When the manager type is *ADMDMN the server is the cluster resource group associated with the cluster administrative domain.

Length of server information
INPUT; BINARY(4)

The length of the server information. The length of the server information parameter must be set to the size of the SRVI0100 structure.

Format of server information
INPUT; CHAR(8)

The format name of the server information. The following format name is allowed:

"Server Information (SRVI0100 Format)" on page 58 The format of the server information when the server is a cluster.

Server defined output
OUTPUT; CHAR(*)

When the manager type is *ADMDMN, the server defined output contains the 16 character request handle. The request handle is a unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter. The API will generate the request handle and return the handle to the caller of the API.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Monitored Resource Entry (EENT0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	CHAR(10)	Monitored resource type
10	A	CHAR(10)	Monitored resource library name
20	14	BINARY(4)	Length of monitored resource name
24	18	CHAR(*)	Monitored resource name

Field Descriptions

Length of monitored resource name. The length in bytes of the monitored resource name.

Monitored resource library name. The name of the monitored resource library. The allowed values are:

Blank Specify a blank monitored resource library name for the monitored resource types *SYSVAL, *ENVVAR, *NETA, or *TCPA.

Monitored resource library name Specify a monitored resource library name for the monitored resource types *USRPRF, *ASPDEV, *JOBD, and *CLS. Monitored resource types of *USRPRF and *ASPDEV must specify QSYS for the monitored resource library name. Library special values, e.g. *CURLIB or *LIBL, are not allowed.

Monitored resource name. The name of the monitored resource which is being monitored by the system.

Monitored resource type. The type of monitored resource entry which is being monitored by the system. The allowed values are:

**USRPRF* User profile.
**JOBD* Job description.
**CLS* Class.
**ASPDEV* Independent auxiliary storage pool (ASP) device description.
**SYSVAL* System value.
**NETA* Network attribute.
**ENVVAR* System environment variable.
**TCPA* TCP/IP attribute.

Attribute Information (ATRI0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Number of attribute entries
4	4	BINARY(4)	Offset to attribute array
Note: The following fields repeat for each attribute monitored by the system.			
		BINARY(4)	Displacement to next attribute entry
		BINARY(4)	Length of attribute name
		CHAR(*)	Attribute name

Field Descriptions

Attribute name. The name of the attribute to be monitored by the system. For monitored resource types of *ENVVAR, *SYSVAL, *NETA and *TCPA, the attribute name is the same as the resource name.

Displacement to next attribute entry. The displacement in bytes from the beginning of this entry to the beginning of the next attribute entry.

Length of attribute name. The length, in bytes, of the attribute name.

Number of attribute entries. The number of attribute entries in the ATRI0100 format. Tbe allowed values are:

-1 Specify -1 to monitor all allowed attributes for a system object monitored resource.
>0 Specify a value greater than 0 to monitor one or more attributes for a system object monitored resource. A value of 1 must be specifed for monitored resource types *SYSVAL, *ENVVAR, *NETA, or *TCPA.

Offset to attribute array. The offset in bytes from the beginning of this structure to the first attribute entry. A value of 0 must be specified when the Number of attribute entries value is equal to -1.

Server Information (SRVI0100 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of server defined output
4	4	CHAR(10)	Cluster name
14	E	CHAR(10)	Cluster administrative domain name
28	1C	CHAR(30)	Results information

Field Descriptions

Cluster administrative domain name. The name of the cluster administrative domain.

Cluster name. The name of the cluster.

Length of server defined output. The length, in bytes, of the server defined output parameter for this API. If the manager type is *ADMDMN, this value must be 16.

Results information. This field identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the “Usage Notes” on page 59 section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the results information are reserved and must be set to hexadecimal zero.

Usage Notes

Adding a Monitored Resource Entry (MRE) in a Cluster Administrative Domain

See Cluster basics for more information about using this API to add monitored resource entries in a cluster administrative domain.

Unless otherwise noted, all character strings are assumed to be in the CCSID of the program that calls the API.

Monitored resources must exist in the system base auxiliary storage pool (ASP).

The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF1999 D	Errors occurred on command.
CPF2204 D	User profile &1 not found.
CPF3C21 E	Format name &1 is not valid.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB47 D	Cluster Resource Services detected an error and may have ended abnormally.
CPFBB48 D	Cluster Resource Services error detected.
CPFBBB9 D	Cluster administrative domain &1 internal error.

Error Messages

Messages that are delivered through the error code parameter are listed here.

Message ID	Error Message Text
CPDAC07 E	System value QRETSVRSEC not set correctly.
CPF1097 E	No authority to change certain network attributes.
CPF1098 E	No authority to class &1 in library &2.
CPF2634 E	Not authorized to object &1.
CPF2697 E	The request did not complete in the time allotted.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFA0AA E	Error occurred while attempting to obtain space.
CPFA98E E	*JOBCTL special authority required to update system-level environment variables.
CPFAA01 E	Attribute &1 for resource &2 not allowed to be monitored.
CPFAA02 E	Resource &1 cannot be added for &2.

Message ID	Error Message Text
CPFB0CE E	&1 special authority is required.
CPFBB02 E	Cluster &1 does not exist.
CPFBB09 E	Cluster node &1 does not exist in cluster &2.
CPFBB0A E	Cluster node &1 in cluster &2 is not active.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB20 E	Cluster partition detected for cluster &1 by cluster node &2.
CPFBB26 E	Cluster Resource Services is not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB46 E	Cluster Resource Services internal error.
CPFBBB5 E	Resource &1 cannot be added for &2.
CPFBBB6 E	Library &1 not allowed for resource &2.

« API introduced: V5R4

Top | "Server Support APIs," on page 1 | APIs by category

Open List Network Server Storage Spaces (QFPADOLD) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Storage space name qualifier	Input	Char(*)
7	Length of storage space name qualifier	Input	Binary(4)
8	Network server description name qualifier	Input	Char(*)
9	Length of network server description name qualifier	Input	Binary(4)
10	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Open List Network Server Storage Spaces (QFPADOLD) API retrieves a list of storage spaces (virtual disk drives) that can be linked to a server. The functions provided by this API are similar to the Work with Network Server Storage Spaces (WRKNWSSTG) command.

Authorities and Locks

Network Server Storage Space Authority
*RX for IFS directory /QFPNWSSTG

*RX for IFS directory /QFPNWSSTG/name

*R for the files in IFS directory /QFPNWSSTG/name

Note: In the above file paths, 'name' is the name of the storage space.

Network Server Storage Space Lock
None

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of the receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 0 bytes.

List information
OUTPUT; CHAR(80)

Information about the list created by this program. For a description of the layout of this parameter, see "Format of List Information" on page 67.

Number of records to return
INPUT; BINARY(4)

The number of records in the list to put into the receiver variable. The value must be 0 or greater.

Format name
INPUT; CHAR(8)

The format of the information returned. You must use the following format name:

DOLD0100 The output storage space list structure. For more information, see "DOLD0100 Format" on page 62.

Storage space name qualifier
INPUT; CHAR(*)

A restriction on the information to be retrieved. This parameter allows the caller to request a subset of the records that normally would be returned on a given format. If this parameter is *ALL, no information qualification is performed. Allowable values are:

**ALL* Information about all storage spaces.
Storage space name Information about the specified storage space name record only.
*Generic storage space name** Information about storage spaces matching the generic storage space name string.

Length of storage space name qualifier
INPUT; BINARY(4)

The length of the storage space name qualifier.

Network server description name qualifier
INPUT; CHAR(*)

A restriction on the information to be retrieved. This parameter allows the caller to request a subset of the records that normally would be returned on a given format. If this parameter is *ALL, no information qualification is performed. Allowable values are:

**ALL* Information about storage spaces for all network server descriptions, including storage spaces that are not linked to a network server description.
Network server description name Information about storage spaces linked to the specified network server description name only.

Length of network server description name qualifier
INPUT; BINARY(4)

The length of the network server description name qualifier.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

DOLD0100 Format

Format DOLD0100 returns storage space information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Displacement to next storage space record
4	4	BINARY(4)	Length of storage space record
8	8	CHAR(10)	Name
18	12	CHAR(50)	Description
68	44	BINARY(4)	Capacity
72	48	BINARY(4)	Percent used
76	4C	BINARY(4)	Disk pool
80	50	BINARY(4)	Storage space format
84	54	BINARY(4)	Storage space formatting status
88	58	BINARY(4)	Access to storage space
92	5C	BINARY(4)	Displacement to list of server entries
96	60	BINARY(4)	Length of server list entries
100	64	BINARY(4)	Number of server list entries
104	68	BINARY(4)	Clustering quorum resource disk
108	6C	CHAR(10)	Disk pool name
118	76	CHAR(2)	Reserved
120	78	BINARY(4)	Displacement to list of extent entries
124	7C	BINARY(4)	Length of extent list entries
128	80	BINARY(4)	Number of extent list entries
»			
132	84	BINARY(4)	Displacement to cluster quorum disk information
136	88	BINARY(4)	Length of cluster quorum disk information
«			

Offset		Type	Field
Dec	Hex		
These fields repeat for each server entry		BINARY(4)	Dynamic link
		BINARY(4)	Link sequence position
		CHAR(8)	Network server description name
		BINARY(4)	Displacement to server name
		BINARY(4)	Length of server name
		BINARY(4)	Link type
		»	
		BINARY(4)	Storage path identifier
		CHAR(10)	Network server host adapter name
		«	
		CHAR(*)	Server name
These fields repeat for each extent entry		BINARY(4)	Extent capacity
		BINARY(4)	Extent free space
		CHAR(16)	Extent format
		CHAR(64)	Volume name
»			
These fields exist when disk is created as a cluster quorum resource disk		CHAR(10)	Cluster name
		CHAR(15)	Cluster domain name
		CHAR(10)	Cluster connection port
		CHAR(15)	Cluster IP address
		CHAR(15)	Cluster subnet mask
«			

Field Descriptions

Access to storage space. The type of access the server has to the storage space data. The possible values follow:

0 Exclusive update - update access; not shared
1 Shared read - read access; storage space shared with other servers
2 Shared update - update access; storage space shared with other servers

Capacity. The size of the storage space, in megabytes (MB).

»

Cluster connection port. The port number that was specified when creating the cluster quorum resource disk.

Cluster domain name. The name of the Windows cluster domain that was specified when creating the cluster quorum resource disk.

Cluster IP address. The internet address of the cluster that was specified when creating the cluster quorum resource disk.

Cluster name. The name of the Windows cluster that was specified when creating the cluster quorum resource disk.

Clustering quorum resource disk. Specifies whether the storage space was created as a cluster quorum resource disk or not.

0 This storage space is not a cluster quorum resource disk.
1 This storage space is a cluster quorum resource disk.

Cluster subnet mask. The subnet mask of the cluster that was specified when creating the cluster quorum resource disk.

Description. The text description of the storage space.

Disk pool. The disk pool that contains the storage space.

1 System disk pool
2-32 User disk pools
33-255 Independent user disk pools

Disk pool name. The device description name for the independent user disk pool that contains the storage space.

Displacement to cluster quorum disk information. The displacement in bytes from the beginning of the record to the cluster quorum disk information. If this disk is not a cluster quorum disk, this value is zero.

Displacement to list of extent entries. The displacement in bytes from the beginning of the record to the list of disk drive extents found on the storage space. If there are no disk drive extents found on this storage space, this value is zero.

Displacement to list of server entries. The displacement in bytes from the beginning of the record to the list of network server descriptions that are linked to the storage space. If there are no network server descriptions linked to the storage space, this value is zero.

Displacement to next storage space record. The displacement in bytes from the beginning of the record to the next storage space record. If this is the last record, this value is zero.

Displacement to server name. The displacement in bytes from the beginning of the server list entry to the server name.

Dynamic link. Whether the storage space was linked to the server dynamically. The possible values follow:

0 The storage space was not linked dynamically.
1 The storage space was linked dynamically.

Extent capacity. The size of the disk drive extent in megabytes (MB).

Extent format. A blank padded Unicode (CCSID 13488) text string describing the file system type formatted on the volume residing on the disk drive extent. Expected value includes but is not limited to one of the following.

FAT	File Asllocation Table (FAT) file system
FAT32	32-bit File Allocation Table (FAT-32) file system
NTFS	Windows NT file system (NTFS)
**NONE*	The area of the disk is unallocated or unformatted.

Extent free space. The size in megabytes (MB) on this disk extent that is not occupied by data.

Length of cluster quorum disk information. The length of the cluster quorum disk information.

Length of extent list entries. The length of each disk drive extent entry.

Length of server list entries. The length of each server entry.

Length of server name. The length of the server name.

Length of storage space record. The length of the data for the current storage space.

Link sequence position. The relative order that storage spaces are linked to the network server description. Storage spaces are presented to the hosted operating system in the order specified by the link sequence position value. There are several types of storage space links. The link types and the corresponding link sequence position ranges are as follows:

Fixed	Links of this type have the **dynamic link** value set to 0. The link sequence position values range from 1 to 16.
Cluster quorum resource	Links of this type have the **clustering quorum resource disk** value set to 1 and the **access to storage space** value set to 2. The link sequence position value is 1.
Cluster shared	Links of this type have the **clustering quorum resource disk** value set to 0 and the **access to storage space** value set to 2. The link sequence position values range from 1 to 15.
Dynamic	Links of this type have the **dynamic link** value set to 1. The link sequence position values range from 1 to 64.

Link type. The type of linking and access mode used for this disk drive. The possible link types are as follow:

1	This disk drive is linked as a fixed disk.
2	This disk drive is linked as a dynamic disk.
3	This disk drive is linked as a cluster quorum resource disk.
4	This disk drive is linked as a cluster shared disk.

Name. The name of the storage space.

Network server description name. The name of the network server description that is linked to the storage space.

Network server host adapter name. The name of the network server host adapter device description that is being used to access this disk drive. This field will be blank if a network server host adapter is not required to access this disk. The following special value can be returned:

**MULTIPATH* Multiple storage paths are defined in the NWSD for accessing this disk drive.

Number of extent list entries. The number of disk drive extent entries. The maximum number of extent entries per disk drive is 32. If more than 32 extent entries exist on the disk drive, the contents of the 32nd extent entry will be a sum of all remaining extent entries.

Number of server list entries. The number of server entries.

Percent used. The percentage of the storage space capacity that is occupied by data.

Reserved. Reserved area for 4-byte boundary alignment.

Server name. The name of the server that is linked to the storage space.

Storage path identifier. This value represents the storage path identifier of the network server host adapter device being used to access this storage.

1 First storage path in NWSD
2 Second storage path in NWSD
3 Third storage path in NWSD
4 Fourth storage path in NWSD
-1 Default storage path
-2 Multipath as defined in NWSD
-3 Storage path not used for accessing disk

Storage space format. The file system format that is used for the storage space. The possible values follow:

0 High performance file system (HPFS)
1 File Allocation Table (FAT) file system
2 NetWare file system
3 AIX file system
4 Windows NT file system (NTFS)
6 32-bit File Allocation Table (FAT-32) file system
7 Open source file system
254 Unknown (the storage space contains an unknown format type)
255 Pending (the storage space is in the process of being created)

Storage space formatting status. Whether the storage space is properly formatted for the specified file system format and ready to use. The possible values follow:

0 Unknown
1 Complete
2 Incomplete

Volume name. A blank padded Unicode (CCSID 13488) text string describing the volume name residing on the disk drive extent.
If the disk contains more than 32 extents, the volume name on the 32nd extent entry will be *OVERFLOW.

Format of List Information

The following table shows the format of the list information parameter. For detailed descriptions of the field in the table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	CHAR(4)	Request handle
12	C	BINARY(4)	Record length
16	10	CHAR(1)	Information complete indicator
17	11	CHAR(13)	Date and time created
30	1E	CHAR(1)	List status indicator
31	1F	CHAR(1)	Reserved
32	20	BINARY(4)	Length of information returned
36	24	BINARY(4)	First record in buffer
40	28	CHAR(40)	Reserved

Field Descriptions

Date and time created. The date and time the list was created. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7 The date, in YYMMDD (year, month, day) format.
8-13 The time of day, in HHMMSS (hours, minutes, seconds) format.

First record in buffer. The number of the first record in the receiver variable.

Information complete indicator. Whether all information requested has been supplied.

I (x'C9') Incomplete information. An interruption caused the list to contain incomplete information about a buffer or buffers.
P (x'D7') Partial and accurate information. Partial information is returned when the maximum space was used and not all of the buffers requested were read.
C x'C3') Complete and accurate information. All the buffers requested were read and returned.

Length of information returned. The size, in bytes, of the information returned in the receiver variable.

List status indicator. The status of the list building. The indicators are:

0 The list building is pending.
1 The list is in the process of being built.

2 The list has been built completely.
3 An error occurred while building the list. The next call to QGYGTLE will cause the error to be signalled to the caller of QGYGTLE.

Record length. The length of each record of information returned. For variable length records, this value is set to 0. For variable length records, you can obtain the length of individual records from the records themselves.

Records returned. The number of records returned in the receiver variable. This is the smallest of the following values:

- The number of records that fit into the receiver variable.
- The number of records in the list.
- The number of records requested.

Request handle. The handle of the request that can be used for subsequent requests of information from the list. The handle is valid until the Close List (QGYCLST) API is called to close the list or until the job ends.

Note: This field should be treated as a hexadecimal field. It should not be converted from one CCSID to another (for example, EBCDIC to ASCII) because doing so could result in an unusable value.

Reserved. Reserved field. Must be set to hexadecimal or binary zero.

Total records. The total number of records available in the list.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF2634 E	Not authorized to object &1.
CPF3C17 E	Error occurred with input data parameter.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C1E E	Required parameter &1 omitted.
CPF3C1F E	Pointer is not on a 16 byte boundary.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF9806 E	Cannot perform function for object &2 in library &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R1

Top | "Server Support APIs," on page 1 | APIs by category

Open List Network Server Users (QFPADOLU) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)

4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Profile type	Input	Char(10)
7	Profile name	Input	Char(10)
8	Windows domain name qualifier	Input	Char(*)
9	Length of Windows domain name qualifier	Input	Binary(4)
10	Windows server name qualifier	Input	Char(*)
11	Length of Windows server name qualifier	Input	Binary(4)
12	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Open List Network Server Users (QFPADOLU) API generates a list of Windows domains and servers, and the associated profiles enrolled to them in the network. The list will be returned grouped by domain or server name, and within the domain or server, by profile name.

Authorities and Locks

Authority to User Profiles in List of Authorized Users
*READ

Note: Only those profiles to which you have *READ authority are returned in the list.

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of the receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 0 bytes.

List information
OUTPUT; CHAR(80)

Information about the list created by this program. For a description of the layout of this parameter, see "Format of List Information" on page 77.

Number of records to return
INPUT; BINARY(4)

The number of records in the list to put into the receiver variable. The value must be 0 or greater.

Format name
INPUT; CHAR(8)

The format of the information returned. You must use the following format name:

"DOLU0100 Record Format" on page 73	Returns records for user and associated group records, or records for group and associated user records.

"DOLU0150 Record Format" on page 73	Returns records for user and associated group records, or records for group and associated user records. The difference between this format and format DOLU0100 is that this format includes the Windows user name in user records. **Note:** The Windows user name could be different than the enrolled i5/OS user name if Enterprise Identity Mapping (EIM) is being used for the enrolled user.
"DOLU0200 Record Format" on page 74	Returns a short version of records for user and associated group records, or records for group and associated user records. **Note:** This format is faster than formats DOLU0100 and DOLU0150.

Profile type
INPUT; CHAR(10)

Whether profile information returned is on a user profile or group profile basis. Values can be:

**USER*	Retrieve information about profiles defined as users on the selected domains or servers. The group information for each group of which the user is a member is also returned.
**GROUP*	Retrieve information about profiles defined as groups on the selected domains or servers. The users belonging to each group are also returned.
**USERBASIC*	Retrieve information about profiles defined as users on the selected domains or servers. All user records are returned, without the associated group records. **Note:** This option is faster than the *USER option.
**GRPBASIC*	Retrieve information about profiles defined as groups on the selected domains or servers. All group records are returned, without the associated user records. **Note:** This option is faster than the *GROUP option.
**ALL*	Retrieve both group and user information for a single domain or server. All group records are returned first, without the associated user records. Then all user records are returned, without the associated group records. » The records will be returned in the *USER format if a specific user profile is requested. «
**NONE*	Retrieve information for domains and servers, but not for users or groups. Records for domains are returned first, followed by records for servers. No user or group records are returned.

Profile name
INPUT; CHAR(10)

The names of the profiles to be included in the list. A specific profile can be specified, or the profile name can be specified with a generic name (such as 'USER*') to retrieve a list of all profiles starting with the specified letters. Allowable values are:

**ALL*	Information about all profiles
Profile name	Information about the specified profile name record only.
*Generic profile name**	Information about profiles matching the generic profile name string.

Usage note: Avoid the use of characters '5B'X, '7B'X, and '7C'X (CCSID 37 characters '$', '#', and '@', respectively) in profile names.

Windows domain name qualifier
INPUT; CHAR(*)

The names of the Windows domains to be included in the list. A specific domain can be specified, or the domain name can be specified with a generic name (such as 'DMN*') to retrieve a list of all domains starting with the specified letters. Allowable values are:

**ALL*	All domains with which users are enrolled are to be included in the list.
**NONE*	No domains are to be included in the list.

**NWSA*	Only the system default domains shown on the Display Network Server Attributes (DSPNWSA) command are to be included in the list.
Domain name	Information about the specified domain only.
*Generic domain name**	Information about domains matching the generic domain name string.

Length of Windows domain name qualifier
INPUT; BINARY(4)

The length of the Windows domain name qualifier.

Windows server name qualifier
INPUT; CHAR(*)

The names of the Windows servers to be included in the list. A specific server can be specified, or the server name can be specified with a generic name (such as 'SRV*') to retrieve a list of all servers starting with the specified letters. Allowable values are:

**ALL*	All servers with which users are enrolled are to be included in the list.
**NONE*	No servers are to be included in the list.
**NWSA*	Only the system default servers shown on the Display Network Server Attributes (DSPNWSA) command are to be included in the list.
Server name	Information about the specified server only.
*Generic server name**	Information about servers matching the generic server name string.

Length of Windows server name qualifier
INPUT; BINARY(4)

The length of the Windows server name qualifier.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

Format of Data Returned

The list data returned for the DOLU0100, DOLU0150 and DOLU0200 formats contain three types of records:

Server records	Contain information about a domain or server.
User records	Contain information about a user profile.
Group records	Contain information about a group profile.

If the profile type requested is *USER, then records are returned as follows:

- Server 1
  - User 1 for server 1
    - Group 1 for user 1
    - Group 2 for user 1
    - Group n for user 1
  - User 2 for server 1
    - Group 1 for user 2
    - Group 2 for user 2
    - Group n for user 2

  - User n for server 1
- Server 2

If the profile type requested is *GROUP, then records are returned as follows:

- Server 1
  - Group 1 for server 1
    - User 1 for group 1
    - User 2 for group 1
    - User n for group 1
  - Group 2 for server 1
    - User 1 for group 2
    - User 2 for group 2
    - User n for group 2
  - Group n for server 1
- Server 2

If the profile type requested is *USERBASIC, then records are returned as follows:

- Server 1
  - User 1 for server 1
  - User 2 for server 1
  - User n for server 1

If the profile type requested is *GRPBASIC, then records are returned as follows:

- Server 1
  - Group 1 for server 1
  - Group 2 for server 1
  - Group n for server 1

If the profile type requested is *ALL, then records are returned as follows:

- Server 1
  - Group 1 for server 1
  - Group 2 for server 1
  - Group n for server 1
  - User 1 for server 1
  - User 2 for server 1
  - User n for server 1

If the profile type requested is *NONE, then records are returned as follows:

- Domain 1
- Domain 2
- Domain n
- Server 1
- Server 2
- Server n

The following tables describe the order and format of the data for each record format in the list. For detailed descriptions of the fields in the tables, see "DOLU0100, DOLU0150, and DOLU0200 Record Field Descriptions" on page 74.

DOLU0100 Record Format

The following table defines the structure of each record returned for format DOLU0100.

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Displacement to next record
4	4	BINARY(4)	Length of record
8	8	BINARY(4)	Record type
12	C	BINARY(4)	Number records to next record same level
16	10	BINARY(4)	Number records next lower level
20	14	BINARY(4)	Displacement to name
24	18	BINARY(4)	Length of name
28	1C	BINARY(4)	Server type
32	20	BINARY(4)	Enrollment reason
36	24	BINARY(4)	Member enrollment
40	28	BINARY(4)	Profile status
44	2C	CHAR(50)	Profile description
94	5E	CHAR(6)	Enrollment error code
100	64	CHAR(10)	Message file
110	6E	CHAR(10)	Primary message file library
120	78	CHAR(7)	Message ID
See note	See note	CHAR(*)	Name
Note: The displacement and length for the name are specified in the corresponding displacement and length variables.			

DOLU0150 Record Format

The following table defines the structure of each record returned for format DOLU0150.

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Displacement to next record
4	4	BINARY(4)	Length of record
8	8	BINARY(4)	Record type
12	C	BINARY(4)	Number records to next record same level
16	10	BINARY(4)	Number records next lower level
20	14	BINARY(4)	Displacement to name
24	18	BINARY(4)	Length of name
28	1C	BINARY(4)	Server type
32	20	BINARY(4)	Enrollment reason
36	24	BINARY(4)	Member enrollment
40	28	BINARY(4)	Profile status
44	2C	CHAR(50)	Profile description
94	5E	CHAR(6)	Enrollment error code

Offset		Type	Field
Dec	Hex		
100	64	CHAR(10)	Message file
110	6E	CHAR(10)	Primary message file library
120	78	CHAR(7)	Message ID
127	7F	CHAR(1)	Reserved
128	80	BINARY(4)	Displacement to Windows name
132	84	BINARY(4)	Length of Windows name
See note	See note	CHAR(*)	Name
See note	See note	CHAR(*)	Windows name
Note: The displacement and length for the name and Windows name are specified in the corresponding displacement and length variables.			

DOLU0200 Record Format

The following table defines the structure of each record returned for format DOLU0200.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Displacement to next record
4	4	BINARY(4)	Length of record
8	8	BINARY(4)	Record type
12	C	BINARY(4)	Number records to next record same level
16	10	BINARY(4)	Number records next lower level
20	14	BINARY(4)	Displacement to name
24	18	BINARY(4)	Length of name
28	1C	BINARY(4)	Server type
32	20	BINARY(4)	Enrollment reason
36	24	BINARY(4)	Member enrollment
40	28	BINARY(4)	Profile status
See note	See note	CHAR(*)	Name
Note: The displacement and length for the name are specified in the corresponding displacement and length variables.			

DOLU0100, DOLU0150, and DOLU0200 Record Field Descriptions

Displacement to name. The displacement in bytes from the beginning of the record to the name.

Displacement to next record. The displacement in bytes from the beginning of the record to the next record. If this is the last record, this value will be zero.

Displacement to Windows name. The displacement in bytes from the beginning of the record to the Windows name. This field is set to zero if the enrolled i5/OS user name and the Windows user name are the same.

Enrollment error code. This value is only specified if the record is a **user** or **group** type record. Normally it is blank, but it will be set to the last enrollment error code that was received while attempting to create, delete, or update a user on a server. This should only be non-blank if the profile is in one of the recovery or delete statuses.

The enrollment error codes are:

239 User enrollment failed because the password can not be retrieved.
240 Password not available for user enrollment create.
241 Invalid Windows profile name.
245 User enrollment failed because the password can not be retrieved.
247 Update of group failed because domain group type cannot be changed.
248 Unable to delete user from Windows domain or server.
249 Unable to delete group from Windows domain or server.
260 User not created because specified name is a reserved name in Windows.
261 Group not created because specified name is a reserved name in Windows.
262 User not deleted because specified name is a reserved name in Windows.
263 Group not deleted because specified name is a reserved name in Windows.
264 Enrollment request cannot complete because domain controller not found.
265 User not created because user template not found.
266 User not created because user name already exists.
267 Create or update of user failed for an unknown reason.
268 Create or update of group failed for an unknown reason.
269 User was created or updated successfully on Windows. The user, however, was not added to the AS400_Users group on Windows.
270 A request to delete the user from Windows failed because the user is a member of the AS400_Permanent_Users group on Windows.
271 A request to delete the group from Windows failed because the group is not empty on Windows.
272 Add user to an enrolled group failed on Windows.
273 Create failed because user or group already exists on Windows.
274 Add user to a group listed in the user template failed on Windows.
275 User successfully enrolled, but not marked as enrolled on Windows.
276 Group successfully enrolled, but not marked as enrolled on Windows.
277 User unenrolled, but still marked as an enrolled user on Windows.
278 Group unenrolled, but still marked as an enrolled group on Windows.
279 Create or update of user failed on Windows; password not valid.
280 Create or update user failed. A network server description (NWSD) associated with the Windows domain or server must be varied on.
281 Enrollment request to local server not valid on domain controller.
282 Service not authorized to manage domain groups and users on the domain.
296 Unable to create home directory for user.
452 More than one EIM identifier defined for i5/OS user.
454 More than one EIM Windows user profile source association defined.
459 EIM processing error for user.

Enrollment reason. This value is specified only if the record is a **user** type record. It indicates why a profile has been selected for enrollment:

1 The user is a member of a group that has enrolled members.
2 The user is specifically enrolled to a domain or server.

Length of name. The length of the name.

Length of record. The length of the data for the current record.

Length of Windows name. The length of the Windows name. This field is set to zero if the enrolled i5/OS user name and the Windows user name are the same.

Member enrollment. This value is specified only if the record is a group type record. It indicates whether group members are enrolled.

0 Group members are not enrolled.
1 Group members are enrolled.

Message file. This value is specified only if the record is a **user** or **group** type record. The message file that contains the message ID for a message that describes the enrollment error code for a failed enrollment operation.

Message ID. This value is specified only if the record is a **user** or **group** type record. The message that describes the enrollment error code for a failed enrollment operation.

Name. The name of a Windows domain or server, or an enrolled i5/OS user or group, depending on the record type.

Usage note: Avoid the use of characters '5B'X, '7B'X, and '7C'X (CCSID 37 characters '$', '#', and '@', respectively) in user and group profile names.

Number records next lower level. The number of sub-records that exist for a record. For example, if the current record is a group record, followed by three user records for that group, then the number of records next lower level would be three.

Number records to next record same level. The number of records to traverse to locate the next record of the same type (server, user, or group). If there are no more records of the same type, this value is zero.

For example, if the current record is a group record, followed by three user records for the group, followed by another group record, then the number of records to next record same level would be four.

Primary message file library. The library that contains the message file. This value is specified only if the record is a user or group type record.

Note: If the message text is retrieved, use *LIBL (library list) as the message file library so that the correct national language version of the message text is retrieved. If the message file returned by this API is not in any of libraries in the library list, then message retrieval operation will fail with message CPF2407 "Message file x in *LIBL not found." If this occurs, then attempt the retrieval operation again, using the primary message file library that is returned by this API.

Profile description. The description for this profile. This value is specified only if the record is a **user** or **group** type record.

Profile status. The current status of a user or group on a server. This value is specified only if the record is a user or group type record. The following values represent the various status values:

1 Profile is enrolled on the server (up-to-date).
2 Enrollment pending on the server.
3 Enrollment retry pending on the server.
4 Enrollment failed.
5 Unenrollment pending on the server.
6 Unenrollment retry pending on the server.
7 Unenrollment failed.

8 Only group members enrolled. The users in the group are enrolled to the servers, but the group itself is not enrolled to the servers. This status is valid only for a group.
9 Profile is enrolled on the server using the i5/OS profile name. An error occured when looking up the Windows user name using Enterprise Identity Mapping (EIM).

Record type. The record type for a record. The values that represent the various types follow:

1 A server type record.
2 A user type record.
3 A group type record.

Note: The first record in the list is always a server type record.

Reserved. Reserved area for 4-byte boundary alignment.

Server type. The type of server represented by the record. This value is specified only if the record is a **server** type record.

1 A Windows domain.
2 A Windows server.

Windows name. The name of the user on Windows. This field is not returned if the enrolled i5/OS user name and the Windows user name are the same.

Note: The Windows user name could be different than the enrolled i5/OS user name if Enterprise Identity Mapping (EIM) is used for the enrolled user.

Format of List Information

The following table shows the format of the list information parameter. For detailed descriptions of the field in the table, see "Field Descriptions" on page 78.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	CHAR(4)	Request handle
12	C	BINARY(4)	Record length
16	10	CHAR(1)	Information complete indicator
17	11	CHAR(13)	Date and time created
30	1E	CHAR(1)	List status indicator
31	1F	CHAR(1)	Reserved
32	20	BINARY(4)	Length of information returned
36	24	BINARY(4)	First record in buffer
40	28	CHAR(40)	Reserved

Field Descriptions

Date and time created. The date and time the list was created. The 13 characters are:

11	Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.
2-7	The date, in YYMMDD (year, month, day) format.
8-13	The time of day, in HHMMSS (hours, minutes, seconds) format.

First record in buffer. The number of the first record in the receiver variable.

Information complete indicator. Whether all information requested has been supplied.

I	Incomplete information. An interruption caused the list to contain incomplete information about a buffer or buffers.
P	Partial and accurate information. Partial information is returned when the maximum space was used and not all of the buffers requested were read.
C	Complete and accurate information. All the buffers requested were read and returned.

Length of information returned. The size, in bytes, of the information returned in the receiver variable.

List status indicator. The status of the list building. The indicators are:

0	The list building is pending.
1	The list is in the process of being built.
2	The list has been built completely.
3	An error occurred while building the list. The next call to QGYGTLE will cause the error to be signalled to the caller of QGYGTLE.

Record length. The length of each record of information returned. For variable length records, this value is set to 0. For variable length records, you can obtain the length of individual records from the records themselves.

Records returned. The number of records returned in the receiver variable. This is the smallest of the following values:

- The number of records that fit into the receiver variable.
- The number of records in the list.
- The number of records requested.

Request handle. The handle of the request that can be used for subsequent requests of information from the list. The handle is valid until the Close List (QGYCLST) API is called to close the list or until the job ends.

Note: This field should be treated as a hexadecimal field. It should not be converted from one CCSID to another (for example, EBCDIC to ASCII) because doing so could result in an unusable value.

Reserved. Reserved field. Must be set to hexadecimal or binary zero.

Total records. The total number of records available in the list.

Error Messages

Message ID	Error Message Text
CPFA314 E	Memory allocation error.
CPF24B4 E	Severe error while addressing parameter list.

Message ID	Error Message Text
CPF2634 E	Not authorized to object &1.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V5R1

Top | "Server Support APIs," on page 1 | APIs by category

Open List Network Servers (QFPADOLS) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	List information	Output	Char(80)
4	Number of records to return	Input	Binary(4)
5	Format name	Input	Char(8)
6	Server name qualifier	Input	Char(*)
7	Length of server name qualifier	Input	Binary(4)
8	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Open List Network Servers (QFPADOLS) API retrieves a list of servers. The functions provided by this API are similar to the Work with Network Server Status (WRKNWSSTS) command.

Authorities and Locks

Network Server Description Authority
*USE

Network Server Description Lock
*EXCLRD

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of the receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of

the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 0 bytes.

List information
OUTPUT; CHAR(80)

Information about the list created by this program. For a description of the layout of this parameter, see "Format of List Information" on page 86.

Number of records to return
INPUT; BINARY(4)

The number of records in the list to put into the receiver variable. The value must be 0 or greater.

Format name
INPUT; CHAR(8)

The format of the information returned. You must use one of the the following format names:

DOLS0100 The output server list structure for Windows servers only. For more information, see "DOLS0100 Format."

DOLS0200 The output server list structure for *WINDOWS, *LINUX, and *AIX NWS server types (*NETWARE servers are excluded) in the hosting partition. For more information, see "DOLS0200 Format" on page 81.

Server name qualifier
INPUT; CHAR(*)

A restriction on the information to be retrieved. This parameter allows the caller to request a subset of the entries that normally would be returned on a given format. If this parameter is *ALL, no information qualification is performed. Allowable values are:

**ALL*	Information about all servers.
Server name	Information about the specified server name entry only.
*Generic server name**	Information about servers matching the generic server name string.

Length of server name qualifier
INPUT; BINARY(4)

The length of the server name qualifier.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

DOLS0100 Format

Format DOLS0100 returns Windows server information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Displacement to next server entry
4	4	BINARY(4)	Length of server entry
8	8	CHAR(8)	Network server description (NWSD) name
16	10	CHAR(1)	Server status

Offset		Type	Field
Dec	Hex		
17	11	CHAR(3)	Reserved
20	14	BINARY(4)	Displacement to server name
24	18	BINARY(4)	Length of server name
28	1C	BINARY(4)	Displacement to domain name
32	20	BINARY(4)	Length of domain name
36	24	BINARY(4)	Displacement to server description
40	28	BINARY(4)	Length of server description
44	2C	BINARY(4)	Displacement to DNS domain name
48	30	BINARY(4)	Length of DNS domain name
52	34	CHAR(10)	Hardware resource name
62	3E	CHAR(4)	Hardware resource type
66	42	CHAR(3)	Hardware resource model
69	45	CHAR(6)	iSeries Integration version
75	4B	CHAR(7)	iSeries Integration service pack
82	52	CHAR(2)	Reserved
84	54	BINARY(4)	Displacement to list of iSeries Integration hot fix entries
88	58	BINARY(4)	Length of an iSeries Integration hot fix entry
92	5C	BINARY(4)	Number of iSeries Integration hot fix entries
96	60	CHAR(10)	Windows version
106	6A	CHAR(10)	Windows build number
116	74	CHAR(128)	Windows service pack level
See note	See note	CHAR(*)	Server name
See note	See note	CHAR(*)	Domain name
See note	See note	CHAR(*)	Server description
See note	See note	CHAR(*)	DNS domain name
These fields repeat for each iSeries Integration hot fix.		CHAR(7)	iSeries Integration hot fix
		CHAR(1)	Reserved
Note: The displacement to the server name, domain name, server description, and DNS domain name and hot fix are specified in the respective displacement variables.			

DOLS0200 Format

Format DOLS0200 returns *WINDOWS, *LINUX, and *AIX Network Server Information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Displacement to next server entry
4	4	BINARY(4)	Length of server entry

Offset		Type	Field
Dec	Hex		
8	8	CHAR(8)	Network server description (NWSD) name
16	10	CHAR(1)	Server status
17	11	CHAR(3)	Reserved
20	14	BINARY(4)	Displacement to server name
24	18	BINARY(4)	Length of server name
28	1C	BINARY(4)	Displacement to domain name
32	20	BINARY(4)	Length of domain name
36	24	BINARY(4)	Displacement to server description
40	28	BINARY(4)	Length of server description
44	2C	BINARY(4)	Displacement to DNS domain name
48	30	BINARY(4)	Length of DNS domain name
52	34	BINARY(4)	Displacement to server operating system description
56	38	BINARY(4)	Length of server operating system description
60	3C	CHAR(10)	Hardware name
70	46	CHAR(24)	Hardware type and model
94	5E	CHAR(12)	Hardware serial number
106	6A	CHAR(10)	Server operating system type
116	74	CHAR(10)	NWSD connection type
126	7E	CHAR(10)	NWSD operating system
136	88	CHAR(6)	Operating system integration version
142	8E	CHAR(7)	Operating system integration service level
149	95	CHAR(10)	Server operating system version
159	9F	CHAR(10)	Server operating system build ID
169	A9	CHAR(128)	Server operating system service level
297	129	CHAR(1)	Server enrollment enabled
298	12A	CHAR(1)	Domain enrollment enabled
299	12B	CHAR(1)	SBMNWSCMD enabled
300	12C	CHAR(1)	Server statistics enabled
301	12D	CHAR(1)	Dynamic unlinking enabled
302	12E	CHAR(1)	Synchronize integration software enabled
303	12F	CHAR(24)	Hardware type and model at install
See note	See note	CHAR(*)	Server name
See note	See note	CHAR(*)	Domain name
See note	See note	CHAR(*)	Server description
See note	See note	CHAR(*)	DNS domain name
See note	See note	CHAR(*)	Server operating system description
Note: The displacement to the server name, domain name, server description, DNS domain name, and server operating system description are specified in the respective displacement variables.			

Field Descriptions

Displacement to DNS domain name. The displacement in bytes from the beginning of the record to the domain name of the Domain Name System (DNS).

Displacement to domain name. The displacement in bytes from the beginning of the record to the domain name.

Displacement to list of iSeries Integration hot fix entries. The displacement to the first entry in the iSeries Integration hot fix list.

Displacement to next server entry. The displacement in bytes from the beginning of the record to the next server entry. If this is the last entry, this value will be zero.

Displacement to server description. The displacement in bytes from the beginning of the record to the server description.

Displacement to server name. The displacement in bytes from the beginning of the record to the server name.

» **Displacement to server operating system description.** The displacement in bytes from the beginning of the record to the server operating system description. «

DNS domain name. The domain name system (DNS) domain name of this server.

» **Domain enrollment enabled.** Indicates if user profiles may be enrolled to the Domain name.

'0'	No, enrollment is not supported to this servers domain name.
'1'	Yes, enrollment is supported to this servers domain name.

«

Domain name. The domain name of this server.

» **Dynamic unlinking enabled.** Indicates if NWS Storage Space dynamic unlinking operations are supported for this server.

'0'	No, dynamic unlinking is not supported on this server.
'1'	Yes, dynamic unlinking is support on this server.

«

» **Hardware name.** The hardware resource name of the input/output adapter (IOA), or the remote system NWSCFG name for iSCSI connected servers. «

Hardware resource model. The hardware resource model of the input/output adapter (IOA) for the server.

Hardware resource name. The hardware resource name of the input/output adapter (IOA) for the server.

Hardware resource type. The hardware resource type of the input/output adapter (IOA) for the server.

» **Hardware serial number.** The serial number of the remote server hardware. When not available the value will be *N. «

» **Hardware type and model.** The remote server hardware type and model, or the hardware resource type input/output adapter (IOA) for the server. **Hardware type and model at install.** The remote server hardware type and model, or the hardware resource type input/output adapter (IOA) used when the server was originally installed. When not available the value will be *N. «

iSeries Integration hot fix. The identifier for an iSeries Integration hot fix that is installed on the server.

iSeries Integration service pack. The identifier for an iSeries Integration service pack that is installed on the server. » When not available the value will be *N. «

iSeries Integration version. The version number of the iSeries Integration product that is installed on the server. The value has the format VxRyMz, where x is the version, y is the release, and z is the modification. » When not available the value will be *N. «

Length of an iSeries Integration hot fix entry. The length of one iSeries Integration hot fix entry.

Length of DNS domain name. The length of this DNS domain name.

Length of domain name. The length of this domain name.

Length of server description. The length of this server description.

Length of server entry. The length of this server entry.

Length of server name. The length of this server name.

» **Length of server operating system description.** The length of this server operating system description. «

Network server description (NWSD) name. The name of the network server description.

Number of iSeries Integration hot fix entries. The number of times the iSeries Integration hot fix entries are repeated. » Hot fix entries are never used, so this field value will always be zero. «

» **NWSD connection type.** The NWSD Server connection value. The possible values are:

**IXSVR*	The NWSD uses an integrated network server connection.
**ISCSI*	The NWSD uses an iSCSI connection.
**GUEST*	The NWSD uses an operating system running in a logical partition.

«

» **NWSD operating system.** The NWSD Server operating system value. The possible values are:

**WIN32*	The network server description is for a 32bit Windows operating system.
**LINUX32*	The network server description is for a 32bit Linux operating system.
**LINUXPPC*	The network server description is for a Linux Power PC operating system.
**AIXPPC*	The network server description is for an AIX Power PC operating system.

«

» **Operating system integration service level.** The identifier for the operating system integration service level that is installed on the server. When not available the value will be *N. «

» **Operating system integration version.** The version number of the operating system integration product that is installed on the server. The value has the format VxRyMz, where x is the version, y is the release, and z is the modification. When not available the value will be *N. «

Reserved. Reserved space.

» **SBMNWSCMD enabled.** Indicates if the SBMNWSCMD CL command is supported to this server.

'0'	No, SBMNWSCMD is not supported to this server.
'1'	Yes, SBMNWSCMD is supported to this server.

«

Server description. The server description.

» **Server enrollment enabled.** Indicates whether user profiles may be enrolled to this server name.

'0'	No, enrollment to the server name is not available.
'1'	Yes, enrollment to the server name is available.

«

Server name. The name of this server. » The operating systems server name is retrieved for *IXSVR and *ISCSI connection types. The NWSD name is returned for *GUEST connection types. «

»

Server operating system build ID. The current build identifier of the server operating system. When not available the value will be *N.

Server operating system description. A description of the operating system on the server. The values is retreived from the remote server. When not available, the value will be *N.

Server operating system service level. The current service level of the operating system on the server. When not available the value will be *N.

Server operating system type. The remote server operating system type. The possible values are:

**WINDOWS*	Windows operating system types.
**LINUX*	Linux operating system types.
**AIX*	AIX operating system types.

Server operating system version. The current version of the operating system on the server. When not available the value will be *N.

Server statistics enabled. Indicates if server statistics are supported for this server.

'0'	No, server statistics are not supported.
'1'	Yes, server statistics are supported.

«

Server status. The status of the server. The possible values follow:

0	Active
1	Inactive
2	Restricted

3 Pending
4 Unknown
5 Ending
6 Failed
7 Partially shut down
»
8 Degraded
«

» **Synchronize integration software enabled.** Indicates if operating system operations are supported to directly update integration software on this server.

'0'	No, integration software synchronization is not supported.
'1'	Yes, integration software synchronization is supported.

«

Windows build number. The current build number of the Windows operating system on the server.

Windows service pack level. The current service pack level of the Windows operating system on the server.

Windows version. The current version of the Windows operating system on the server.

Format of List Information

The following table shows the format of the list information parameter. For detailed descriptions of the field in the table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	CHAR(4)	Request handle
12	C	BINARY(4)	Record length
16	10	CHAR(1)	Information complete indicator
17	11	CHAR(13)	Date and time created
30	1E	CHAR(1)	List status indicator
31	1F	CHAR(1)	Reserved
32	20	BINARY(4)	Length of information returned
36	24	BINARY(4)	First record in buffer
40	28	CHAR(40)	Reserved

Field Descriptions

Date and time created. The date and time the list was created. The 13 characters are:

1 Century, where 0 indicates years 19*xx* and 1 indicates years 20*xx*.

2-7 The date, in YYMMDD (year, month, day) format.
8-13 The time of day, in HHMMSS (hours, minutes, seconds) format.

First record in buffer. The number of the first record in the receiver variable.

Information complete indicator. Whether all information requested has been supplied.

I Incomplete information. An interruption caused the list to contain incomplete information about a buffer or buffers.
P Partial and accurate information. Partial information is returned when the maximum space was used and not all of the buffers requested were read.
C Complete and accurate information. All the buffers requested were read and returned.

Length of information returned. The size, in bytes, of the information returned in the receiver variable.

List status indicator. The status of the list building. The indicators are:

0 The list building is pending.
1 The list is in the process of being built.
2 The list has been built completely.
3 An error occurred while building the list. The next call to QGYGTLE will cause the error to be signalled to the caller of QGYGTLE.

Record length. The length of each record of information returned. For variable length records, this value is set to 0. For variable length records, you can obtain the length of individual records from the records themselves.

Records returned. The number of records returned in the receiver variable. This is the smallest of the following values:

- The number of records that fit into the receiver variable.
- The number of records in the list.
- The number of records requested.

Request handle. The handle of the request that can be used for subsequent requests of information from the list. The handle is valid until the Close List (QGYCLST) API is called to close the list or until the job ends.

Note: This field should be treated as a hexadecimal field. It should not be converted from one CCSID to another (for example, EBCDIC to ASCII) because doing so could result in an unusable value.

Reserved. Reserved field. Must be set to hexadecimal or binary zero.

Total records. The total number of records available in the list.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF2625 E	Not able to allocate object &1.
CPF2634 E	Not authorized to object &1.
CPF3C17 E	Error occurred with input data parameter.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C1D E	Length specified in parameter &1 not valid.

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C1F E	Pointer is not on a 16 byte boundary.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF9806 E	Cannot perform function for object &2 in library &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

Note: CPF2634 and CPF2625 messages are returned in the error code parameter, and are not signaled.

API introduced: V4R5

Top | "Server Support APIs," on page 1 | APIs by category

Process Network Server User (QFPADPNU) API

Required Parameter Group:

1	Option	Input	Char(10)
2	Profile type	Input	Char(10)
3	Profile name	Input	Char(10)
4	Windows domain name	Input	Char(*)
5	Length of Windows domain name	Input	Binary(4)
6	Windows server name	Input	Char(*)
7	Length of Windows server name	Input	Binary(4)
8	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Process Network Server User (QFPADPNU) API performs an error recovery operation on a selected user or group enrollment entry for a particular Windows domain or server.

Enrollment and unenrollment requests can fail for various reasons. If an enrollment or unenrollment request has failed, the administrator can correct the error and then use this API with the *RETRY option to retry the failed enrollment or unenrollment request. If an unenrollment request has failed and the error cannot be corrected, the administrator can use this API with the *REMOVE option to remove the user or group from the list of enrolled users or groups for a domain or server.

Authorities and Locks

User profile
*READ

Required Parameter Group

Option
INPUT; CHAR(10)

The type of operation to be performed. You must use one of the following options:

**REMOVE* This option removes the entry from the list of users or groups that are enrolled to a Windows domain or server. This option is only valid for those entries that are in the process of being unenrolled (status of unenrollment pending, unenrollment retry pending, or unenrollment failed).
For example, if an unenrollment request failed because the server that the user was enrolled to no longer exists, then the administrator could use the *REMOVE option of this API to remove the user from the enrollment list for that server.

**RETRY* This option immediately retries the enrollment request to the Windows domain or server. This option is normally used after an enrollment or unenrollment request has failed and the reason for the failure has been corrected.
For example, if an enrollment request failed because the user template does not exist on the server, then the administrator could create the user template and then use the *RETRY option of this API to reattempt the enrollment request.

Profile type
INPUT; CHAR(10)

Whether profile information is for a user profile or group profile. Values can be:

**USER* The profile information is for a user profile on the selected domain or server.
**GROUP* The profile information is for a group profile on the selected domain or server

Profile name
INPUT; CHAR(10)

The name of the profile to process. Allowable values are:

Profile name Perform the operation for the specified user or group profile.

Usage note: Avoid the use of characters '5B'X, '7B'X, and '7C'X (CCSID 37 characters '$', '#', and '@', respectively) in profile names.

Windows domain name
INPUT; CHAR(*)

The name of the Windows domain to process. Allowable values are:

**NONE* No domain is to be processed.
Domain name Perform the operation for the specified domain.

Length of Windows domain name
INPUT; BINARY(4)

The length of the Windows domain name.

Windows server name
INPUT; CHAR(*)

The name of the Windows server to process. Allowable values are:

**NONE* No server are to be processed.
Server name Perform the operation for the specified server.

Length of Windows server name
INPUT; BINARY(4)

The length of the Windows server name.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF2634 E	Not authorized to object &1.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V5R1

Top | "Server Support APIs," on page 1 | APIs by category

Remove Monitored Resource Entry (QfpadRmvMonitoredResourceEntry) API

Required Parameter Group:

1	Manager type	Input	Char(10)
2	Monitored resource information	Input	Char(*)
3	Length of monitored resource information	Input	Binary(4)
4	Format of monitored resource information	Input	Char(8)
5	Server information	Input	Char(*)
6	Length of server information	Input	Binary(4)
7	Format of server information	Input	Char(8)
8	Server defined output	Output	Char(*)
9	Error code	I/O	Char(*)

Service Program Name: QFPADAP1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Remove Monitored Resource Entry (QfpadRmvMonitoredResourceEntry) API removes a Monitored Resource Entry (MRE) from the monitored resource directory. A monitored resource may be a system object or a set of attributes not associated with a specific system object, such as the set of system environment variables. Monitored resource entries are added using the Add Monitored Resource Entry (QfpadAddMonitoredResourceEntry) API.

This API will do the following when the manager type is *ADMDMN:

- Removes the monitored resource entries on all nodes in the cluster administrative domain.
- Once the monitored resource entry is removed, changes to the monitored resource will not be synchronized within the cluster administrative domain.
- Any system objects that were created or any system environment variables that were added when a monitored resource was added to the monitored resource directory will not be removed. The user must manually delete any resources that are no longer needed.

This API requires the following if the manager type is *ADMDMN:

- When removing monitored resource entries from a cluster administrative domain, the cluster administrative domain must exist and all nodes in the domain must be active.
- Resources and attributes cannot be removed from the monitored resource directory when the cluster administration domain is partitioned.
- The program that calls this API must be running under a user profile that exists on every node in the cluster administrative domain.
- This API operates in an asynchronous mode. See Behavior of Cluster Resource Services APIs for more information.

Authorities and Locks

The following authorities are explicitly checked for by this API. Additional authority checking may be performed by APIs or CL commands called as a result of this API. See the Monitored resources topic in the information center for more information.

System configuration (*IOSYSCFG) special authority is required to use this API.

Configuration Object Authority (applies to a configuration object for which the MRE is removed)
*OBJMGT

Library Authority
*EXECUTE

Request Information User Queue Authority
*OBJOPR, *ADD

Request Information User Queue Library Authority
*EXECUTE

Request Information User Queue Lock
*EXCLRD

Required Parameter Group

Manager type
INPUT; CHAR(10)

The management function which is to monitor the resource. *ADMDMN must be specified.

Monitored resource information
INPUT; CHAR(*)

The monitored resource information provided as input.

Length of monitored resource information
INPUT; BINARY(4)

The length of the monitored resource information.

Format of monitored resource information
INPUT; CHAR(8)

The format name of the monitored resource information. The following format name is allowed:

"Monitored Resource Entry (EENT0100 Format)" on page 92 The format of monitored resource information in a cluster administrative domain.

Server information
INPUT; CHAR(*)

The information about the server where the monitored resource will be monitored by the system. When the manager type is *ADMDMN, the server is the cluster resource group associated with the cluster administrative domain.

Length of server information
INPUT; BINARY(4)

The length of the server information. The length of the server information parameter must be set to the size of the SRVI0100 structure.

Format of server information
INPUT; CHAR(8)

The format name of the server information. The following format name is allowed:

"Server Information (SRVI0100 Format)" on page 93 The format of the server information when the server is a cluster.

Server defined output
OUTPUT; CHAR(*)

When the manager type is *ADMDMN, the server defined output contains the 16 character request handle. The request handle is a unique string or handle that identifies this API call. It is used to associate this call to any responses placed on the user queue specified in the results information parameter. The API will generate the request handle and return the handle to the caller of the API.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Input Parameter Section

The following information is returned in the input parameter section. For detailed descriptions of the fields in this table, see "Field Descriptions."

Monitored Resource Entry (EENT0100 Format)

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Monitored resource type
10	A	CHAR(10)	Monitored resource library name
20	14	BINARY(4)	Length of monitored resource name
24	18	CHAR(*)	Monitored resource name

Field Descriptions

Length of monitored resource name. The length in bytes of the monitored resource name.

Monitored resource library name. The name of the monitored resource library. The allowed values are:

Blank — Specify a blank monitored resource library name for the monitored resource entry types *SYSVAL, *ENVVAR, *NETA, or *TCPA.

Monitored resource library name — Specify a monitored resource library name for the monitored resource entry types *USRPRF, *ASPDEV, *JOBD, and *CLS. Monitored resource types of *USRPRF and *ASPDEV must specify QSYS for the monitored resource library name. Library special values, e.g. *CURLIB or *LIBL, are not allowed.

Monitored resource name. The name of the monitored resource.

Monitored resource type. The type of monitored resource. The allowed values are:

**USRPRF*	User profile.
**JOBD*	Job description.
**CLS*	Class.
**ASPDEV*	Independent auxiliary storage pool (ASP) device description.
**SYSVAL*	System value.
**NETA*	Network attribute.
**ENVVAR*	System environment variable.
**TCPA*	TCP/IP attribute.

Server Information (SRVI0100 Format)

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Length of server defined output
4	4	CHAR(10)	Cluster name
14	E	CHAR(10)	Cluster administrative domain name
24	18	CHAR(30)	Results information

Field Descriptions

Cluster administrative domain name. The name of the cluster administrative domain.

Cluster name. The name of the cluster.

Length of server defined output. The length, in bytes, of the server defined output parameter for this API. If the manager type is *ADMDMN, this value must be 16.

Results information. This field identifies a qualified user queue field and is followed by a reserved field.

Qualified user queue: Completion information is returned to this user queue, which exists on the node from which the API was called, after the function has completed. See the “Usage Notes” section of this API for a description of the data that is placed on this queue. This is a 20-character field. The first 10 characters contain the user queue name, and the second 10 characters contain the user queue library name. No special values are supported. QTEMP, *LIBL, *CURLIB are not valid library names. The attributes of this user queue must be keyed.

Reserved: The last 10 characters of the results information are reserved and must be set to hexadecimal zero.

Usage Notes

Removing a Monitored Resource Entry (MRE) in a Cluster Administrative Domain

See Cluster basics for more information about using this API to remove a monitored resource entry in a cluster administrative domain.

Unless otherwise specified, all character strings are assumed to be in the CCSID of the program that calls this API.

The following identifies the data sent to the user queue (excluding the message text).

Message ID	Message Text
CPCBB01 C	Cluster Resource Services API &1 completed.
CPF2204 D	User profile &1 not found.
CPF3C21 D	Format name &1 is not valid.
CPF3CF2 D	Error(s) occurred during running of &1 API.
CPFBB17 D	&1 API cannot be processed in cluster &2.
CPFBB2D D	Timeout detected while waiting for a response.
CPFBB35 D	The user profile name &1 is not valid for this request.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB47 D	Cluster Resource Services detected an error and may have ended abnormally.
CPFBB48 D	Cluster Resource Services error detected.
CPFBBA1 D	All cluster command user queues busy.
CPFBBB9 D	Cluster administrative domain &1 internal error.

Error Messages

Messages that are delivered through the error code parameter are listed here. The data (messages) sent to the results information user queue are listed in the "Usage Notes" on page 93 above.

Message ID	Error Message Text
CPF1098 E	No authority to class &1 in library &2.
CPF2634 E	Not authorized to object &1.
CPF2697 E	The request did not complete in the time allotted.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFAA03 E	Resource &1 in library &4 type &3 cannot be removed for managed resource type &2.
CPFB0CE E	&1 special authority is required.
CPFBB02 E	Cluster &1 does not exist.
CPFBB0F E	Cluster resource group &1 does not exist in cluster &2.
CPFBB20 E	Cluster partition detected for cluster &1 by cluster node &2.
CPFBB26 E	Cluster Resource Services is not active or not responding.
CPFBB32 E	Attributes of user queue &1 in library &1 are not valid.
CPFBB39 E	Current user does not have IOSYSCFG special authority.
CPFBB46 E	Cluster Resource Services internal error.
CPFBBB6 E	Library &1 not allowed for resource &2.

Retrieve Monitored Resource Information (QfpadRtvMonitoredResourceInfo) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format of receiver variable	Input	Char(8)
4	Manager type	Input	Char(10)
5	Monitored resource information	Input	Char(*)
6	Length of monitored resource information	Input	Binary(4)
7	Format of monitored resource information	Input	Char(8)
8	Server information	Input	Char(*)
9	Length of server information	Input	Binary(4)
10	Format of server information	Input	Char(8)
11	Server defined output	Output	Char(*)
12	Error code	I/O	Char(*)

Service Program Name: QFPADAP1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Retrieve Monitored Resource Information (QfpadRtvMonitoredResourceInfo) API returns information about monitored resources.

A monitored resource may be a system object or a set of attributes not associated with a specific system object, such as the set of system environment variables. System objects and attributes are monitored by the system to allow notifications of changes to the attributes to be passed to a supporting system function or application for system management purposes.

This API will do the following when the manager type is *ADMDMN:

- Return information about monitored resource entries that is stored on the system where the API was called. If cluster resource services is not started or the cluster administrative domain is partitioned, the information returned may not reflect the most current information in the cluster administrative domain.

Authorities and Locks

The following authorities are checked for by this API before returning information about the following monitored resource types.

User Profile Authority
*READ

Job Description Authority
*USE

Class Authority
*USE

Configuration Object Authority
*USE

Library Authority
*EXECUTE

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable to hold the monitored resource entry information.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable in bytes. The minimum length is 8 bytes. If the length of the receiver variable is not large enough to hold the repeating fields of the information specified in the Monitored Resource Entry Information Format Name parameter, only the number of records specified by the Number of Records Returned field will be returned.

Format of receiver variable
INPUT; CHAR(8)

The format name of the monitored resource entry information which will be returned. The following format names are allowed:

"Monitored Resource Entry Information (DENR0100 Format)" on page 97 Information about one or more monitored resource entries.

"Attribute Information (DENR0200 Format)" on page 98 Information about the monitored attributes for a single monitored resource.

Manager type
INPUT; CHAR(10)

The management function which is to monitor the resource. *ADMDMN must be specified.

Monitored resource information
INPUT; CHAR(*)

Information to identify the monitored resource or resources that are to be returned. See EENT0100 for a description of the monitored resource information.

Length of monitored resource information
INPUT; BINARY(4)

The length of the monitored resource information.

Format of monitored resource input information
INPUT; CHAR(8)

The format name of the monitored resource information. The following format name is allowed:

"Monitored Resource Input Information (EENT0100 Format)" on page 102 The format of monitored resource input information in a cluster administrative domain.

Server information
INPUT; CHAR(*)

Information about the server. When the manager type is *ADMDMN, the server is the cluster resource group associated with the cluster administrative domain.

Length of server information
INPUT; BINARY(4)

The length of the server information. The length of the server information parameter must be set to the size of the SRVI0100 structure.

Format of server information
INPUT; CHAR(8)

The format name of the server information. The following format name is allowed:

"Server Information (SRVI0100 Format)" on page 103 The format of the server information when the server is a cluster.

Server defined output
OUTPUT; CHAR(*)

Information returned which is specific to the server type. When the manager type is *ADMDMN, the following information is returned:

Information status.

One byte which indicates the consistency of the retrieved information.

0 The information is consistent for all active nodes in the cluster administrative domain.
1 The information retrieved from the node running the API may not be consistent with all active nodes in the cluster. In order to obtain consistent information:

Call this API on an active node in the cluster, if the node running the API is not active.
Start Cluster Resource Services on the node running the API if it is not active.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Monitored Resource Entry Information (DENR0100 Format)

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to first record
12	C	BINARY(4)	Length of fixed portion of record
16	10	BINARY(4)	Number of records returned
Note: Record entry information. These fields are repeated for each record entry returned.			
		BINARY(4)	Displacement to next record
		CHAR(10)	Monitored resource type
		CHAR(10)	Monitored resource library name
		BINARY(4)	Global status
		BINARY(4)	Resource status
		BINARY(4)	Displacement to monitored resource name
		BINARY(4)	Length of monitored resource name
		BINARY(4)	Displacement to node array
		BINARY(4)	Length of node array entry
		BINARY(4)	Number of node array entries

Offset		Type	Field
Dec	Hex		
		BINARY(4)	Displacement to message information
		BINARY(4)	Length of message information
Note: Use the displacements in the fixed portion of the record to get to the following fields.			
		CHAR(*)	Monitored resource name
		CHAR(*)	Node array
		CHAR(*)	Message information

Attribute Information (DENR0200 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to first record
12	C	BINARY(4)	Length of fixed portion of record
16	10	BINARY(4)	Number of records returned
Note: Record entry information. These fields are repeated for each record entry returned.			
		BINARY(4)	Displacement to next record
		CHAR(10)	Monitored resource type
		CHAR(10)	Monitored resource library name
		BINARY(4)	Global status
		BINARY(4)	Resource status
		BINARY(4)	Displacement to monitored resource name
		BINARY(4)	Length of monitored resource name
		BINARY(4)	Displacement to attribute list
		BINARY(4)	Length of fixed portion of attribute list entry
		BINARY(4)	Number of attribute list entries
Note: Use the displacements in the fixed portion of the record to get to the following fields.			
		CHAR(*)	Monitored resource name
		CHAR(*)	Attribute list

Field Descriptions

Attribute list. A list of attributes for the monitored resource described in the current record. This structure consists of a list of repeating entries. See “Attribute List Entry Format” on page 101 for the format of a single entry.

Bytes available. The number of bytes of data available to be returned to the user. If all data is returned, this is the same as the number of bytes returned. If the receiver variable was not big enough to contain all of the data, this is the number of bytes that can be returned.

Bytes returned. The number of bytes of data returned to the user. This is the lesser of the number of bytes available to be returned or the length of the receiver variable. If the requested monitored resource is not monitored by the system, this value will be set to 20.

Displacement to attribute list. The displacement in bytes from the start of the current entry to the attribute list.

Displacement to message information. The displacement in bytes from the start of the current entry to the message information.

Displacement to monitored resource name. The displacement in bytes from the start of the current entry to the monitored resource name field.

Displacement to next record. The displacement in bytes from the start of the current entry to the next entry.

Displacement to node array. The displacement in bytes from the start of the current entry to the node array. This will be set to 0 if the global status for the monitored resource is not set to 1 (INCONSISTENT).

Global status. The status of the resource across the active cluster administrative domain. Possible values are:

0	*CONSISTENT*	The values for all the resource's attributes monitored by the system are the same within the cluster administrative domain.
1	*INCONSISTENT*	One or more of the monitored attributes for a monitored resource entry are not set to the value known by the cluster administrative domain on one or more nodes in the domain.
2	*PENDING*	The values of the monitored attributes are in the process of being synchronized across the cluster administrative domain.
3	*ADDED*	The monitored resource entry and its attributes have been added to the monitored resource directory in the cluster administrative domain but has not yet been synchronized.

Length of fixed portion of attribute list entry. The length, in bytes, of the fixed fields in an attribute list entry.

Length of fixed portion of record. The length of the fixed fields in the format description. If no records are returned, this field will be set to 0.

Length of message information. The length, in bytes, of the message information.

Length of monitored resource name. The length, in bytes, of the monitored resource entry name.

Length of node array entry. The length, in bytes, of the node array entry.

Message information. The message information associated with the monitored resource or attribute. See "Message Information Format" on page 102 for the format of this data.

Monitored resource name. The name of the monitored resource entry for which information is returned.

Monitored resource library name. The name of the library of the monitored resource. Blanks will be returned for the monitored resource types *SYSVAL, *NETA, *ENVVAR, *TCPA, and *ALL.

Monitored resource type. The type of monitored resource. Possible values are:

**USRPRF*	User profile
**JOBD*	Job description

**CLS*	Class
**ASPDEV*	Independent auxiliary storage pool (ASP) device description
**SYSVAL*	System value
**NETA*	Network attributes
**ENVVAR*	System environment variable
**TCPA*	TCP/IP attributes
**ALL*	Return information for all monitored resource types

Node array. The array of cluster node names that represents the list of nodes where a problem caused the global status of the monitored resource to be set to INCONSISTENT.

Number of attribute list entries. The number of entries in the attribute list. If the receiver variable is not large enough to hold all of the information, this number contains only the number of attribute list entries actually returned.

Number of node array entries. The number of entries in the node array.

Number of records returned. The number of entries in the DENR0100 or DENR0200 format returned. If the receiver variable is not large enough to hold all of the information, this number contains only the number of record entries actually returned. If no records are returned, this field is set to 0.

Offset to first record. The offset, in bytes, from the beginning of the receiver variable to the first record. If no entries are returned, the offset is set to zero.

Reserved. This field will be set to hexadecimal zeroes.

Resource status. The resource status of the monitored resource on the local node.The following values are valid:

0	*CURRENT*	There are no pending updates for the monitored resource on this node.
2	*DLTPND*	The monitored resource has been deleted on a node in the cluster and the process has not fully completed for all nodes.
3	*UPDPND*	There are pending updates for the monitored resource on this node.
4	*RSTPND*	The monitored resource has been restored on a node in the cluster and the process has not fully completed for all nodes.
5	*RNMPND*	The monitored resource has been renamed on a node in the cluster and the process has not fully completed for all nodes.
6	*MOVPND*	The monitored resource has been moved on a node in the cluster and the process has not fully completed for all nodes.
8	*DLTFAIL*	The resource on has been deleted on a node in the cluster and the administrator needs to complete the process.
9	*UPDFAIL*	An update of the resource on this node has failed.
10	*RSTFAIL*	The resource has been restored on a node in the cluster and the administrator needs to complete the process.
11	*RNMFAIL*	The resource has been renamed on a node in the cluster and the administrator needs to complete the process.
12	*MOVFAIL*	The resource has been moved on a node in the cluster and the administrator needs to complete the process.

Attribute List Entry Format

For detailed descriptions of the fields in this table, see "Field Descriptions."

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Displacement to next attribute list entry
4	4	BINARY(4)	Global status
8	8	BINARY(4)	Data type
12	C	BINARY(4)	Displacement to attribute name
16	10	BINARY(4)	Length of attribute name
20	14	BINARY(4)	Displacement to attribute value
24	18	BINARY(4)	Length of attribute value
		CHAR(*)	Attribute name
		CHAR(*)	Attribute value

Field Descriptions

Attribute name. The name of the attribute name.

Attribute value. The value of the attribute. If the global status for the resource is CONSISTENT, this is the value of the attribute across the active cluster administrative domain. If the global status is INCONSISTENT or PENDING, this value represents the last attempted change to the attribute within the active cluster administrative domain.

Data Type. The data type of the attribute value. Valid data types are:

0 Character data
1 Binary data
2 Object data
3 Authority data
4 User profile options data
5 Supplemental groups data
6 Path data
7 Message log data
8 Text data
9 Environment variable data

Displacement to attribute name. Displacement from the beginning of the current attribute list entry to the attribute name.

Displacement to attribute value. The displacement from the beginning of the current attribute list entry to the value.

Displacement to next attribute list entry. Displacement from the beginning of the current attribute list entry to the next attribute list entry. The value will be zero, if there are no more entries in the list.

Global status. The status of the resource attribute value across the active cluster administrative domain. Possible values are:

0 *CONSISTENT* The value for this attribute is the same on all active nodes in the cluster administrative domain.

1 *INCONSISTENT* The value for this attribute is not the same on all active nodes in the cluster administrative domain.

Length of attribute name. The length in bytes of the attribute name.

Length of attribute value. The length in bytes of the attribute value.

Message Information Format

For detailed descriptions of the fields in this table, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(7)	Message ID
7	7	CHAR(10)	Message file
17	11	CHAR(10)	Message library
27	1B	CHAR(1)	Reserved
28	1C	BINARY(4)	Displacement to message replacement data
32	20	BINARY(4)	Length of message replacement data
36	24	BINARY(4)	Message replacement data CCSID
40	28	CHAR(*)	Message replacement data

Message information is available when the global status is INCONSISTENT.

Field Descriptions

Displacement to message replacement data. The displacement in bytes from the beginning of the message information to the message replacement data. If there is no message information data, this field is set to zero.

Length of message replacement data. The length, in bytes, of the message replacement data.

Message file. The message file.

Message ID. The message identifier.

Message library. The message library.

Message replacement data. The message replacement data.

Message replacement data CCSID. The coded character set identifier (CCSID) of the convertible character (CCHAR) portion of the exception data.

Reserved. Reserved field. This field will be set to hexadecimal zeroes.

Monitored Resource Input Information (EENT0100 Format)

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(10)	Monitored resource input type
10	A	CHAR(10)	Monitored resource input library name

Offset		Type	Field
Dec	Hex		
20	14	BINARY(4)	Length of monitored resource input name
24	18	CHAR(*)	Monitored resource input name

Field Descriptions

Length of monitored resource input name. The length, in bytes, of the monitored resource name.

Monitored resource input library name. The name of the library for the specified monitored resource. Allowed values are:

Blank	Specify a blank monitored resource library name for the monitored resource types *SYSVAL, *ENVVAR, *NETA, *TCPA, or *ALL.
Monitored resource library name	The name of the library for monitored resource types *USRPRF, *ASPDEV, *JOBD, and *CLS. Monitored resource types of *USRPRF and *ASPDEV must specify QSYS for the monitored resource library name. Library special values,e.g. *CURLIB or *LIBL, are not allowed.

Monitored resource input name. The name of the monitored resource or list of resources for which information is returned. The allowed values are:

Monitored resource name	The name of a monitored resource.
**ALL*	Returns information for all monitored resources.

Monitored resource input type. The type of monitored resource for which information is to be returned. Possible values are:

**USRPRF*	User profile
**SBSD*	Subsystem description
**JOBD*	Job description
**CLS*	Class
**ASPDEV*	Independent auxiliary storage pool (ASP) device description
**SYSVAL*	System value
**NETA*	Network attributes
**ENVVAR*	System environment variable
**TCPA*	TCP/IP attributes
**ALL*	Return information for all monitored resource types

Server Information (SRVI0100 Format)

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of server defined output
4	4	CHAR(10)	Cluster name
14	D	CHAR(10)	Cluster administrative domain name
24	18	CHAR(30)	Reserved

Field Descriptions

Cluster administrative domain name. The name of the cluster administrative domain where the monitored resource is monitored by the system.

Cluster name. The name of the cluster associated with the cluster administrative domain.

Length of server defined output. The length, in bytes, of the server defined output parameter. For a *ADMDMN manager type, this value must be 1.

Reserved. This field will be set to hexadecimal zeroes.

Usage Notes

Retrieving Monitored Resource Information in a Cluster Administrative Domain

See Cluster basics for more information about using this API to retrieve monitored resource information in a cluster administrative domain.

Unless otherwise noted, all character strings are assumed to be in the CCSID of the program that calls this API.

Error Messages

Message ID	Error Message Text
CPF1098 E	No authority to class &1 in library &2.
CPF2634 E	Not authorized to object &1.
CPF2697 E	The request did not complete in the time allotted.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C29 E	Object name &1 is not valid.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Not authorized to object &2 in &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9820 E	Not authorized to use library &1.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPFA0AA E	Error occurred while attempting to obtain space.
CPFB0CE E	&1 special authority is required.
CPFBBB6 E	Library &1 not allowed for resource &2.

API introduced: V5R4

Top | "Server Support APIs," on page 1 | APIs by category

Retrieve Network Server Information (QFPADRNI) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	NWSD server name	Input	Char(8)
5	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

» The Retrieve Network Server Information (QFPADRNI) API retrieves information from a Windows or Intel Linux server. The amount of information returned depends on the size of the variable. The information returned is similar to the information returned by the Work with Network Server Status (WRKNWSSTS) command's option 5, Display details. «

Authorities and Locks

Network Server Description Authority
*USE

Network Server Description Lock
*EXCLRD

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The variable to receive the network server information. You can specify the size of this area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data that the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable. If the length is larger than the size of the receiver variable, the results may not be predictable. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The format of the network server information to be returned. The valid format names are:

»

DRNI0100 Returns information about the Windows or Intel Linux server version and the iSeries Integration product version that is installed on the server.
DRNI0200 Returns information about the Windows or Intel Linux server's current status.

«

NWSD server name
INPUT; CHAR(8)

The name of the NWSD for which to retrieve information. The name must be specified in upper case characters and padded with blanks to a full 8 characters.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

DRNI0100 Format

The DRNI0100 format returns network server version information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Server operating system version
18	12	CHAR(10)	Server operating system build ID
28	1C	CHAR(128)	Server operating system service level
156	9C	CHAR(6)	iSeries Integration version
162	A2	CHAR(38)	Reserved
200	C8	BINARY(4)	Offset to list of iSeries Integration service pack entries
204	CC	BINARY(4)	Length of an iSeries Integration service pack entry
208	D0	BINARY(4)	Number of iSeries Integration service pack entries
212	D4	BINARY(4)	Offset to list of iSeries Integration hot fix entries
216	D8	BINARY(4)	Length of an iSeries Integration hot fix entry
220	DC	BINARY(4)	Number of iSeries Integration hot fix entries
These fields repeat for each iSeries Integration service pack.		CHAR(7)	iSeries Integration service pack
		CHAR(1)	Reserved
These fields repeat for each iSeries Integration hot fix.		CHAR(7)	iSeries Integration hot fix
		CHAR(1)	Reserved

DRNI0200 Format

The DRNI0200 format returns network server status information.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Connected users
12	C	BINARY(4)	Processor used
16	10	BINARY(4)	Paging file used
20	14	BINARY(4)	Registry quota used
24	18	BINARY(4)	Offset to list of multiple processor usage entries
28	1C	BINARY(4)	Length of a multiple processor usage entry
32	20	BINARY(4)	Number of multiple processor usage entries
This field repeats for each multiple processor usage entry		BINARY(4)	Multiple processor usage

Field Descriptions

Bytes available. The length of all data available for the requested format. All available data is returned if enough space is provided.

Bytes returned. The length of all data actually returned. If the data is truncated because the receiver variable is not large enough to hold the data, this value is less than the bytes available.

Connected users. Number of user connections currently present on this server.

iSeries Integration hot fix. The identifier for an iSeries Integration hot fix that is installed on the server.

iSeries Integration service pack. The identifier for an iSeries Integration service pack that is installed on the server.

iSeries Integration version. The version number of the iSeries Integration product that is installed on the server. The value has the format VxRyMz, where x is the version, y is the release, and z is the modification.

Length of a multiple processor usage entry. The length of one multiple processor usage entry.

Length of an iSeries Integration hot fix entry. The length of one iSeries Integration hot fix entry.

Length of an iSeries Integration service pack entry. The length of one iSeries Integration service pack entry.

Multiple processor usage. The percent usage for servers that support multiple processors. This number is the processor usage value for an individual processor, represented as an integer value between 0 and 100. A usage value is returned for each processor, so the number of values returned depends on the number of processors on the server.

Number of iSeries Integration hot fix entries. The number of times the iSeries Integration hot fix entries are repeated.

Number of iSeries Integration service pack entries. The number of times the iSeries Integration service pack entries are repeated.

Number of multiple processor usage entries. The number of times the multiple processor usage entries are repeated.

Offset to list of iSeries Integration hot fix entries. The offset to the first entry in the iSeries Integration hot fix list.

Offset to list of iSeries Integration service pack entries. The offset to the first entry in the iSeries Integration service pack list.

Offset to list of multiple processor usage entries. The offset to the first entry in the multiple processor usage list.

Paging file used. The percentage of the server paging file used. This percentage is expressed as an integer value between 0 and 100.

Processor used. The percentage of the server's processor used. This percentage is expressed as an integer value between 0 and 100.

If the server supports a single processor, this number is the usage for the individual processor. If the server supports multiple processors, this number is the average processor usage for all of the processors.

Registry quota used. The percentage of the server's registry used. This percentage is expressed as an integer value between 0 and 100.

Reserved. Reserved space.

Server operating system build ID. The current build identifier of the server operating system.

Server operating system service level. The service level of the operating system that is installed on the server.

Server operating system version. The current version of the operating system on the server.

Error Messages

Message ID	Error Message Text
CPF0C4A E	Product record not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF2625 E	Not able to allocate object &1.
CPF2634 E	Not authorized to object &1.
CPF3C17 E	Error occurred with input data parameter.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF9806 E	Cannot perform function for object &2 in library &3.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R5

Top | "Server Support APIs," on page 1 | APIs by category

Retrieve Network Server User Attributes (QFPADRUA) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Profile name	Input	Char(10)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

The Retrieve Network Server User Attributes (QFPADRUA) API retrieves the network server attributes for a user or a group. The functions provided by this API are similar to the Display Network Server User Attributes (DSPNWSUSRA) command.

Authorities and Locks

User profile
*READ

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The receiver variable that receives the information requested. You can specify the size of the area to be smaller than the format requested as long as you specify the length parameter correctly. As a result, the API returns only the data the area can hold.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable provided. The length of the receiver variable parameter may be specified up to the size of the receiver variable specified in the user program. If the length of the receiver variable parameter specified is larger than the allocated size of the receiver variable specified in the user program, the results are not predictable. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The format of the information returned. Indicates whether this call is interested in the user characteristics of the profile or the group characteristics of the profile. Profiles can behave as both user and group profiles, and unique enrollment characteristics can be set for each aspect of the profile. You must use one of the following format names:

DRUA0100 The output network server user attributes structure. For more information, see "DRUA0100 Format" on page 110.
DRUA0200 The output network server group attributes structure. For more information, see "DRUA0200 Format" on page 110.

Profile name
INPUT; CHAR(10)

The name of the user or group profile whose network server attributes are to be retrieved. The values can be:

**NWSA* The system default network server attributes defined by the Change Network Server Attributes (CHGNWSA) command are retrieved. This value is allowed only when format DRUA0100 is specified.
Name The network server attributes for the specified user or group profile are retrieved.

Usage note: Avoid the use of characters '5B'X, '7B'X, and '7C'X (CCSID 37 characters '$', '#', and '@', respectively) in profile names.

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter. If this parameter is omitted, diagnostic and escape messages are issued to the application.

Format of the Retrieved Information

The formats of the information retrieved by this API are listed below. The caller should pass a receiver variable large enough to handle most requests. On return, the bytes available field of the receiver variable will contain the required size of the receiver variable to hold all information from the request. If this is greater than the original receiver variable size passed in, then the caller must allocate a larger receiver variable before retrying the operation.

DRUA0100 Format

The following table shows the format of the network server user attributes.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(12)	Reserved
20	14	BINARY(4)	Offset Windows domain list
24	18	BINARY(4)	Length Windows domain entries
28	1C	BINARY(4)	Number Windows domains
32	20	BINARY(4)	Offset Windows server list
36	24	BINARY(4)	Length Windows server entries
40	28	BINARY(4)	Number Windows servers
See note	See note	See "Format of Windows domain information" on page 112	Windows domain list
See note	See note	See "Format of Windows server information" on page 113	Windows server list
Note: The offsets to the Windows domain list and Windows server list are specified in the offset Windows domain list and offset Windows server list variables, respectively.			

DRUA0200 Format

The following table show the format of the network server group attributes.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(10)	Group member enrollment
18	12	CHAR(2)	Reserved
20	14	BINARY(4)	Offset Windows domain list
24	18	BINARY(4)	Length Windows domain entries
28	1C	BINARY(4)	Number Windows domains
32	20	BINARY(4)	Offset Windows server list

Offset		Type	Field
Dec	Hex		
36	24	BINARY(4)	Length Windows server entries
40	28	BINARY(4)	Number Windows servers
See note	See note	See "Format of Windows domain information" on page 112	Windows domain list
See note	See note	See "Format of Windows server information" on page 113	Windows server list
Note: The offsets to the Windows domain list and Windows server list are specified in the offset Windows domain list and offset Windows server list variables, respectively.			

DRUA0100 and DRUA0200 Field Descriptions

Bytes available. The length of all data available for the requested format. All available data is returned if enough space is provided.

Bytes returned. The length of all data actually returned. If the data is truncated because the receiver variable is not large enough to hold the data, this value is less than the bytes available.

Group member enrollment. This field indicates whether or not the group members (users in the group) are enrolled to the server. This field is only returned for format DRUA0200. The following are the valid values for this field:

**ALL*	The group and the users in the group are enrolled to the servers.
**MBRONLY*	Only the users in the group are enrolled to the servers. The group itself is not enrolled to the servers.
**NONE*	Only the group is enrolled to the servers. The users in the group are not enrolled to the servers.

Length Windows domain entries. The length in bytes of each element in the list of Windows domains.

Length Windows server entries. The length in bytes of each element in the list of Windows servers.

Number Windows domains. The number of entries in the list of Windows domain names. If this value is 0, then there are no Windows domains in the list.

Number Windows servers. The number of entries in the list of Windows server names. If this value is 0, then there are no Windows servers in the list.

Offset Windows domain list. The offset from the start of the receiver variable to the list of Windows domain entries. A value of zero indicates there are no list entries. See "Format of Windows domain information" on page 112 for details of the individual entries.

Offset Windows server list. The offset from the start of the receiver variable to the list of Windows server entries. A value of zero indicates there are no list entries. See "Format of Windows server information" on page 113 for details of the individual entries.

Reserved. Reserved space.

Windows domain list. The list of Windows domain entries. See "Format of Windows domain information" for details of the individual entries.

Windows server list. The list of Windows server entries. See "Format of Windows server information" on page 113 for details of the individual entries.

Format of Windows domain information

The following table describes the layout of the information returned to describe a Windows domain. To locate the first such entry, add the receiver variable start to the value in 'Offset Windows domain list' noted above. Each succeeding entry can be located by adding the 'Length Windows domain entries' size to the address of the current entry.

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Windows group type
10	A	CHAR(2)	Reserved
12	C	BINARY(4)	Displacement to Windows domain name
16	10	BINARY(4)	Length of Windows domain name
20	14	BINARY(4)	Displacement to Windows user template name
24	18	BINARY(4)	Length of Windows user template name
See Note	See Note	CHAR(*)	Windows domain name
See Note	See Note	CHAR(*)	Windows user template name
Note: The displacement to the Windows domain name and Windows user template name are specified in the displacement to Windows domain name and displacement to Windows user template name variables, respectively.			

Windows Domain Field Descriptions

Displacement to Windows domain name. The displacement in bytes from the beginning of the entry to the Windows domain name.

Displacement to Windows user template name. The displacement in bytes from the beginning of the entry to the Windows user template name.

Length of Windows domain name. The length of the Windows domain name.

Length of Windows user template name. The length of the Windows user template name.

Reserved. Reserved space.

Windows domain name. This is the name of a domain in which the user is enrolled. The values can be:

**NWSA* The user is enrolled to the system default domains that are defined by the Change Network Server Attributes (CHGNWSA) command.

Name The specific name of the domain in which the user is enrolled.

Windows group type. This is the type of group created on a Windows domain.

The values can be:

**GLOBAL* A global group was created.

**LOCAL* A local group was created.

blank Blanks are returned if retrieving user attributes.

Windows user template name. The name of a Windows user that was used as a template when creating the specified user on Windows. The values can be:

**NONE* No user template was used when creating the specified user.
Name The specific name of the user template that was used when creating the specified user.

Format of Windows server information

The following table describes the layout of the information returned to describe a Windows server. To locate the first such entry, add the receiver variable start to the value in 'Offset Windows server list' noted above. Each succeeding entry can be located by adding the 'Length Windows server entries' size to the address of the current entry.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Displacement to Windows server name
4	4	BINARY(4)	Length of Windows server name
8	8	BINARY(4)	Displacement to Windows user template name
12	C	BINARY(4)	Length of Windows user template name
See Note	See Note	CHAR(*)	Windows server name
See Note	See Note	CHAR(*)	Windows user template name
Note: The displacement to the Windows server name and Windows user template name are specified in the displacement to Windows server name and displacement to Windows user template name variables, respectively.			

Windows Server Field Descriptions

Displacement to Windows server name. The displacement in bytes from the beginning of the entry to the Windows server name.

Displacement to Windows user template name. The displacement in bytes from the beginning of the entry to the Windows user template name.

Length of Windows server name. The length of the Windows server name.

Length of Windows user template name. The length of the Windows user template name.

Windows server name. The name of a server in which the user is enrolled. The values can be:

**NWSA* The user is enrolled to the system default servers that are defined by the Change Network Server Attributes (CHGNWSA) command.
Name The specific name of the server in which the user is enrolled.

Windows user template name. The name of a Windows user that was used as a template when creating the specified user on Windows. The values can be:

**NONE* No user template was used when creating the specified user.
Name The specific name of the user template that was used when creating the specified user.

Error Messages

Message ID	Error Message Text
CPF2204 E	User profile &1 not found.
CPF24B4 E	Severe error while addressing parameter list.
CPF2634 E	Not authorized to object &1.
CPF3C19 E	Error occurred with receiver variable specified.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3CF1 E	Error code parameter not valid.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.
CPF9999 E	Function check. &1 unmonitored by &2 at statement &5, instruction &3.

API introduced: V5R1

Top | "Server Support APIs," on page 1 | APIs by category

Retrieve Remote Network Server Status (QITDRSTS) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Remote system network server configuration name	Input	Char(10)
5	Timeout	Input	Binary(4)
6	Error Code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

Retrieve Remote Network Server Status (QITDRSTS) API retrieves status information about a remote network server.

Authorities and Locks

Remote System Network Server Configuration Authority
*USE

Required Parameter Group

Receiver variable
OUTPUT; CHAR(*)

The variable that is to receive the remote network server status information.

Length of receiver variable
INPUT; BINARY(4)

The length of the area that is referred to by the receiver variable parameter. If the amount of information to be returned is greater than this value, the information will be truncated to this length. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The content and format of the status information that is returned for the remote network server. The possible format name is:

RSTS0100 Basic network server information

See "Format of Remote Network Server Status Information" for a description of these formats.

Remote system network server configuration name
INPUT; CHAR(10)

The name of the remote system network server configuration to retrieve status for.

Timeout
INPUT; BINARY(4)

The number of seconds to allow the remote system to respond to the status request. Allowed range is 30-300 seconds

Error code
I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see Error Code Parameter.

Format of Remote Network Server Status Information

Specifying RSTS0100 will return the basic status information.

For detailed descriptions of the fields returned in these formats, see "Field Descriptions."

RSTS0100 Format

Use this format to find out basic status information about the remote network server.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Remote network server status code

Field Descriptions

Bytes available. The length of all data available to return. All available data is returned if enough space is provided.

Bytes returned. The length of all data actually returned. If the data is truncated because the receiver variable was not sufficiently large to hold all the data available, this value will be less than the bytes available.

Remote network server status code. The status of the remote network server. The level of status that can be returned is dependent on the type of service processor. Some service processors only return power on (32)/power off (0) status while others return more granular information. Status codes include:

0 System powered off or state unknown to service processor
1 In POST portion of boot process access.
2 Stopped in POST due to an error
3 Booted flash or a system partition

4 Booting operating system * (see note below)
5 In operating system * (see note below)
6 CPU held in reset
7 System Power On before POST
32 System powered on

*** Note: If the operating system or application code does not update the system state to 'In operating system', the system state will never change from 'Booting operating system' to 'In operating system' state.**

Error Messages

Message ID	Error Message Text
CPF2114 E	Cannot allocate object &1 in &2 type *&3.
CPF2189 E	Not authorized to object &1 in &2 type *&3.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C24 E	Length of the receiver variable is not valid.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3CF4 E	Severe error occurred during API processing. Reason code &1.
CPF96CB E	Network server configuration &1 not found.
CPF96CD E	Network server configuration type &2 is not valid.
CPFA31E E	Required parameter &1 omitted.
CPFB752 E	Internal error in &2 API.
CPFC401 E	Remote server or enclosure not found.
CPFC402 E	Remote server or enclosure not found.
CPFC403 E	Remote server or enclosure not found.
CPFC404 E	More than one remote server or enclosure found for the serial number.
CPFC405 E	Remote server configuration error.
CPFC406 E	Program error has occurred.
CPFC408 E	Secure Socket Layer (SSL) connection to the service processor failed.
CPFC409 E	Not authorized to connect to the service processor on the remote server.
CPFC40B E	The certificate of the Service processor is expired.
CPFC40C E	Remote server configuration error.
CPFC40E E	Cannot connect to IBM Director Server.
CPFC40F E	Cannot connect to the service processor of the remote server or enclosure.

« API introduced: V5R4

Top | "Server Support APIs," on page 1 | APIs by category

Dynamic Host Configuration Protocol Server Exit Programs

The iSeries(TM) Dynamic Host Configuration Protocol (DHCP) server assigns and releases TCP/IP addresses from client hosts in a network. Exit points have been provided so that user-written programs can be called from the running DHCP server. They allow for customer-supplied security validation of incoming client requests, as well as notification when IP addresses are assigned or released.

The DHCP Request Packet Validation exit program allows for incoming client request packets to be screened by a user program that starts any processing on it. The user program is able to communicate back to the DHCP server that either the packet should be discarded or that it should be allowed to process.

The Address Binding Notification exit program and the Address Release Notification exit program are for notification purposes only. These allow for user applications to keep track of which IP addresses are currently in use and by which clients.

The Dynamic Host Configuration Protocol Server exit programs are:

- "DHCP Address Binding Notification Exit Program" allows for notification each time the Dynamic Host Configuration Protocol (DHCP) server assigns an IP address to a specific host.
- "DHCP Address Release Notification Exit Program" on page 119 allows for notification each time the DHCP server releases an IP address from its specific client host assignment binding.
- "DHCP Request Packet Validation Exit Program" on page 120 provides additional control for restricting which incoming DHCP and Bootstrap Protocol (BOOTP) message request packets from client hosts are processed by their DHCP server, and which are rejected.

Top | "Server Support APIs," on page 1 | APIs by category

Host Server Exit Programs

Host server 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. For information about how the host server exit programs can be used and how to configure them, as well as sample programs, see Using server exit programs.

The Host server exit programs are:

- Central server
- Data queue server
- Database server
- File Server
- Network print server
- Remote command and distributed program call server
- Signon server

Top | "Server Support APIs," on page 1 | APIs by category

Exit Programs

These are the Exit Programs for this category.

DHCP Address Binding Notification Exit Program

Required Parameter Group:

1	Request type	Input	Binary(4)
2	Client IP address	Input	Char(*)
3	Length of IP address	Input	Binary(4)
4	Client identifier	Input	Char(*)
5	Length of client identifier	Input	Binary(4)
6	Lease duration	Input	Binary(4)
7	Response packet	Input	Char(*)
8	Length of response packet	Input	Binary(4)

Exit Point Name: QIBM_QTOD_DHCP_ABND
Exit Point Format Name: DHCA0100

The DHCP Address Binding Notification exit program allows for notification each time the DHCP server assigns an IP address to a specific client host. This is known as address binding. The exit program will be called any time the DHCP server sends an acknowledgement packet (known as a DHCP ACK) to a DHCP client or sends a Bootstrap Protocol (BOOTP) reply packet to a BOOTP client. The two possible client IP address binding packets follow:

1. A DHCP ACK from the server tells a DHCP client that it has been granted a lease to use a specific IP address for a specific period of time. Acknowledgements are sent when a DHCP client accepts an initial offer of an IP address made by the server, as well as each time that a DHCP client asks for the renewal of an existing IP address lease and the server grants it.
2. A BOOTP reply packet is sent from the server in response to a BOOTP request packet from a client if the server decides to assign an IP address to that client. The BOOTP reply notifies the BOOTP client what IP address it has been assigned. Unlike DHCP, the BOOTP protocol does not support the concept of temporary leasing. Therefore, the address binding to the BOOTP client is considered permanent and there will be no ongoing exchange of renewal bindings. The DHCP server internally treats a BOOTP client binding as an infinite lease.

When an exit program is added to the exit point, it is called immediately after the DHCP server has transmitted one of the client IP address binding packets described above. This is for notification purposes only, and no data is expected to be sent back to the DHCP server from the exit program. The exit program will be sent information about the protocol type (DHCP or BOOTP), the identity of the client, the IP address, and the duration of the lease. In addition, a copy of the actual packet that was transmitted will be sent.

Note: Since this is an exit point of the DHCP server, the exit program can only be used to obtain notification of address bindings for the BOOTP client if the DHCP server is running. It cannot be used to obtain notification of IP address assignments for the BOOTP client made by the iSeries BOOTP server.

Authorities and Locks

None.

Required Parameter Group

Request type
INPUT; BINARY(4)

The type of request protocol being used between the client and server. The possible values are:

1 DHCP
2 BOOTP

Client IP address
INPUT; CHAR(*)

The Internet Protocol (IP) address that was bound to the client host. This string is in dotted decimal format and left-justified.

Length of client IP address
INPUT; BINARY(4)

The length (in bytes) of the client IP address.

Client identifier
INPUT; CHAR(*)

The unique identifier of the client that the IP address has been bound to. This is usually the hardware address of the client machine.

Length of client identifier
INPUT; BINARY(4)

The length (in bytes) of the client identifier string.

Lease duration
INPUT; BINARY(4)

The duration (in seconds) of the lease period for which the client may now use the IP address.

This field should be treated as a 32-bit unsigned number. The special value follows:

hex FFFFFFFF If all 32 bits are set to 1, this indicates that the lease is an infinite lease, which does not expire.

Response packet
INPUT; CHAR(*)

This is the DHCP or BOOTP message response packet that was transmitted from the DHCP server to the client host just prior to this notification exit program being called.

The formats of the packets are defined and maintained by the Internet Engineering Task Force (IETF) standards body. Refer to the following IETF Request For Comments (RFC) documents for the specifications:

- RFC 951, Bootstrap Protocol (BOOTP)
- RFC 1542, Clarifications and Extensions for the Bootstrap Protocol
- RFC 2131, Dynamic Host Configuration Protocol
- RFC 2132, DHCP Options and BOOTP Vendor Extensions

Note: Since the packet is presented to the exit program just as it was transmitted on the network, it should be noted that any data areas of the packet that are defined as type string or character by the RFCs will be US-ASCII. On the iSeries, it is recommended that this data be treated as CCSID 819.

Length of response packet
INPUT; BINARY(4)

The length (in bytes) of the response packet.

Exit program introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

DHCP Address Release Notification Exit Program

Required Parameter Group:

1	Reason for release	Input	Binary(4)
2	Client IP address	Input	Char(*)
3	Length of client IP address	Input	Binary(4)
4	Client identifier	Input	Char(*)
5	Length of client identifier	Input	Binary(4)

Exit Point Name: QIBM_QTOD_DHCP_ARLS
Exit Point Format Name: DHCR0100

The DHCP Address Release Notification exit program allows for notification each time the DHCP server releases an IP address from its specific client host assignment binding.

An address could have been released from the client that it was bound to for one of the following reasons:

1. The client explicitly sent a DHCP RELEASE request to the DHCP server.

2. The lease duration that the client was last given for using the IP address has expired.
3. A DHCP server administrator has explicitly released the IP address.

When an exit program is added to the exit point, it is called whenever the DHCP server releases an IP address from the specific client host that it had been assigned to. This is for notification purposes only, and no data is expected to be sent back to the DHCP server from the exit program. The exit program will be sent information about the IP address, the identity of the client, and the reason for the IP address being released.

Authorities and Locks

None.

Required Parameter Group

Reason for release
INPUT; BINARY(4)

A reason that the DHCP server has released the IP address from being bound to the specific client host. The possible values are:

1 The DHCP server received a DHCP RELEASE packet from the client.
2 The duration of the lease that the client was last given for use of the IP address has expired.
3 The release is forced by DHCP server administrator.

Client IP address
INPUT; CHAR(*)

The Internet Protocol (IP) address that was just released from its client host binding. This string is in dotted decimal format and left-justified.

Length of client IP address
INPUT; BINARY(4)

The length (in bytes) of the client IP address.

Client identifier
INPUT; CHAR(*)

The unique identifier of the client from which the IP address has been released. This is usually the hardware address of the client machine.

Length of client identifier
INPUT; BINARY(4)

The length (in bytes) of the client identifier string.

Exit program introduced: V4R2

Top | "Server Support APIs," on page 1 | APIs by category

DHCP Request Packet Validation Exit Program

Required Parameter Group:

1	Request packet	Input	Char(*)
2	Length of request packet	Input	Binary(4)
3	Allow operation	Output	Binary(4)

Exit Point Name: QIBM_QTOD_DHCP_REQ
Exit Point Format Name: DHCV0100

The Dynamic Host Configuration Protocol (DHCP) Request Packet Validation exit program provides additional control for restricting which incoming DHCP and Bootstrap Protocol (BOOTP) message request packets from client hosts are processed by the DHCP server, and which should be rejected. Any restrictions that are imposed by the exit program are in addition to any validation that is performed by the DHCP server program through its configuration options. When an exit program is added to the exit point, it is called by the DHCP server each time an incoming DHCP or BOOTP request packet is received, but before any processing of the packet has taken place. The packet data is passed to the exit program just as it was received off the network without any additional formatting or data conversion. The exit program sets the allow operation parameter to indicate whether the DHCP server should continue with normal processing of the request packet. If the exit program indicates that the request packet should be rejected, the DHCP server will discard the packet without any further processing.

Note: Since this is an exit point of the DHCP server, the exit program can only be used to restrict BOOTP packets if the DHCP server is running. It cannot be used to restrict BOOTP packets from being processed by the BOOTP server.

Authorities and Locks

None.

Required Parameter Group

Request packet
INPUT; CHAR(*)

The DHCP or BOOTP message request packet that was initiated from some client host and received by the DHCP server. It is in its initial form, which is how it comes off of the network.

The formats of the packets are defined and maintained by the Internet Engineering Task Force (IETF) standards body. Refer to the following IETF Request For Comments (RFC) documents for the specifications:

- RFC 951, Bootstrap Protocol (BOOTP)
- RFC 1542, Clarifications and Extensions for the Bootstrap Protocol
- RFC 2131, Dynamic Host Configuration Protocol
- RFC 2132, DHCP Options and BOOTP Vendor Extensions

Note: Since the packet is presented to the exit program just as it was received from the network, it should be noted that any data areas of the packet that are defined as type string or character by the RFCs will be US-ASCII. On the iSeries, it is recommended that this data be treated as CCSID 819.

Length of request packet
INPUT; BINARY(4)

The length (in bytes) of the request packet.

Allow operation
OUTPUT; BINARY(4)

Whether the DHCP server should continue processing the request packet, or whether it should be rejected. The possible values are:

0 The request packet should be rejected
1 Processing of the request packet should be allowed to continue

Exit program introduced: V4R2

Top | “Server Support APIs,” on page 1 | APIs by category

Appendix. Notices

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

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This Application Programming Interfaces (API) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
Advanced 36
Advanced Function Printing
Advanced Peer-to-Peer Networking
AFP
AIX
AS/400
COBOL/400
CUA
DB2
DB2 Universal Database
Distributed Relational Database Architecture
Domino
DPI
DRDA
eServer
GDDM
IBM
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
i5/OS
iSeries
Lotus Notes
MVS
Netfinity
Net.Data
NetView
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
PowerPC
PrintManager
Print Services Facility
RISC System/6000
RPG/400
RS/6000
SAA
SecureWay
System/36
System/370
System/38
System/390
VisualAge
WebSphere
xSeries

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.

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