

Client Management Support APIs (V5R2)

Table of Contents

- [Client Management Support APIs](#)
- APIs
 - [Add Client](#) (QZCAADDC, QzcaAddClient)
 - [Get Client Handle](#) (QzcaGetClientHandle)
 - [Refresh Client Information](#) (QZCAREFC, QzcaRefreshClientInfo)
 - [Remove Client](#) (QZCARMVC, QzcaRemoveClient)
 - [Update Client Information](#) (QZCAUPDC, QzcaUpdateClientInfo)
- Exit programs
 - [Add Client](#)
 - [Lotus Notes Database](#)
 - [Refresh Client Information](#)
 - [Remove Client](#)
 - [Update Client Information](#)

Client Management Support APIs

The Client Management Support APIs and the related exit programs are provided to act on the client information database found on the iSeries. Functions include adding clients to and removing clients from the server, updating client information, and refreshing the database with directory, hardware, and software information stored on the client.

The **client information database** is a set of physical and logical files enabling the use of iSeries database interfaces or SQL to retrieve client information. The information stored in the database has been retrieved using the Simple Network Management Protocol (SNMP) GET and GETNEXT requests from the client when connected to the iSeries.

The Client Software Management and Configuration APIs can call an exit program if it has been registered with the associated exit point of that API. The exit program can be registered by using the Work with Registration Information (WRKREGINF) command. When the API function is complete, the Add, Refresh, and Update exit programs are called; the associated exit programs notify you when the API functions have been completed. The Remove exit program is called before the API performs the remove function.

The Client Software Management and Configuration APIs are:

- [Add Client](#) (QZCAADDC, QzcaAddClient) allows applications to make clients known to iSeries.
- [Get Client Handle](#) (QzcaGetClientHandle) allows applications to know the assigned client handle if the client ID is known.
- [Refresh Client Information](#) (QZCAREFC, QzcaRefreshClientInfo) allows applications to refresh hardware, software, or all the information stored on the system with the information stored at the client.
- [Remove Client](#) (QZCARMVC, QzcaRemoveClient) allows an application to delete a client from the client database on the system, thereby making a client unknown to the system.
- [Update Client Information](#) (QZCAUPDC, QzcaUpdateClientInfo) allows applications to update miscellaneous information about the client that is stored on the system.

The Client Software Management and Configuration exit programs are:

- [Add Client](#) allows applications to be notified when clients are made known to the system.
- [Lotus Notes Database](#) processes an SQL table of all changes that occurred in the Notes database that is being shadowed from DB2 UDB for iSeries.
- [Refresh Client Information](#) allows applications to be notified when information about clients has been refreshed through the Refresh Client Information API from the system.
- [Remove Client](#) allows applications to be notified when clients are removed from the system.
- [Update Client Information](#) allows applications to be notified when information about clients has been updated through the Update Client Information API from the system.

Add Client (QZCAADDC, QzcaAddClient) API

Required Parameter Group:

1	Client handle assigned	Output	Char(12)
2	Client handle requested	Input	Char(12)
3	Client ID	Input	Char(*)
4	Community name	Input	Char(*)
5	Connectivity information	Input	Char(*)
6	Error code	I/O	Char(*)

Default Public Authority: *EXCLUDE

Service Program: QZCACLT

Threadsafe: No

The Add Client (OPM, QZCAADDC; ILE, QzcaAddClient) API allows applications to make clients known to the server. Use the Refresh Client Information API to retrieve more information about the client.

New clients have a system status of unknown until the Update Client Information (QZCAUPDC, QzcaUpdateClientInfo) API is used to indicate *ACTIVE or *INACTIVE.

Authorities and Locks

API Public Authority

*EXCLUDE

Required Parameter Group

Client handle assigned

OUTPUT; CHAR(12)

A unique key to identify this managed client. This parameter returns either the value of the client handle requested if one was specified, or a generated value for this client if one was not specified.

Client handle requested

INPUT; CHAR(12)

A unique key to identify this managed client. For ILE, this parameter can be omitted. If it is omitted a unique key is generated by this API. A NULL-terminated string is required if it is specified. For OPM, this is a required parameter. If this parameter is less than 12 bytes, it is padded with hexadecimal zeros.

Client ID

INPUT; CHAR(*)

An administratively assigned system name for this managed client. By convention, this is the client's fully qualified domain name. A NULL-terminated string is required for this parameter. The length of client ID is 1 through 255.

Community name

INPUT; CHAR(*)

A relationship between an SNMP agent and a set of SNMP managers that is used for authentication, access-control, and proxy. A NULL-terminated string is required for this parameter. The length of the community name is 1 through 255.

Connectivity information

INPUT; CHAR(*)

The connection information used to contact the client. For the format of this parameter, see [Format for Connectivity Information Parameter](#).

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see [Error Code Parameter](#).

Format for Connectivity Information Parameter

The connectivity information parameter has the following format. For detailed descriptions of the fields in the table, see [Field Descriptions](#).

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	Node type
1	1	CHAR(3)	Reserved
4	4	BINARY(4)	Length of node name
8	8	CHAR(*)	Node name

Field Descriptions

Length of node name. The length of the node name.

Valid values follow:

Node Type *Description*

3 The length is 7 through 15

Node name. The name of the node. The format is dependent on the node type.

Valid values follow:

Node Type *Description*

3 An internet address. The internet address is specified in the form nnn.nnn.nnn.nnn, where nnn is an integer ranging from 0 through 255.

Note: An internet address of 000.000.000.000 or 255.255.255.255 is treated as an invalid address.

Node type. The type of connection.

The following values are valid:

Node Type *Description*

3 Host address

Reserved. An ignored field.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF5E81 E	Client ID already exists.
CPF5E82 E	Client handle already exists.
CPF5E84 E	Node type not valid.
CPF5E85 E	Node name length not valid.
CPF5E88 E	Parameter length not valid.
CPF5E95 E	Internal processing error.
CPF96B8 E	Internet address &1 not valid.
CPF9845 E	Error occurred while opening file &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Get Client Handle (QzcaGetClientHandle) API

Required Parameter Group:

1	Client handle	Output	Char(12)
2	Client ID	Input	Char(*)
3	Error code	I/O	Char(*)

Service Program: QZCACL T

Default Public Authority: *EXCLUDE

Threadsafe: No

The Get Client Handle (QzcaGetClientHandle) API allows applications to know the client handle assigned if the client ID is known.

Authorities and Locks

API Public Authority

*EXCLUDE

Required Parameter Group

Client handle

OUTPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Client ID

INPUT; CHAR(*)

An administratively assigned system name for this managed client. By convention, this is the client's fully qualified domain name. A NULL-terminated string is required for this string. The length of the client ID is 1 through 255.

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
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5E88 E	Parameter length not valid.
CPF5E95 E	Internal processing error.
CPF5E96 E	Client ID does not exist.
CPF9845 E	Error occurred while opening file &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Refresh Client Information (QZCAREFC, QzcaRefreshClientInfo) API

Required Parameter Group:

1	Client handle	Input	Char(12)
2	Information type	Input	Char(10)
3	Time out	Input	Binary(4)
4	Error code	I/O	Char(*)

Service Program: QZCACL T

Default Public Authority: *EXCLUDE

Threadsafe: No

The Refresh Client Information (OPM, QZCAREFC; ILE, QzcaRefreshClientInfo) API allows applications to refresh hardware, software, or all information stored on the server with the information stored at the client.

Authorities and Locks

API Public Authority

*EXCLUDE

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Information type

INPUT; CHAR(10)

The information that should be refreshed at the server.

**ALL* Client details, hardware, and software information should be refreshed at the server.

**HDW* Only hardware information needs to be refreshed at the server.

**SFW* Only software information needs to be refreshed at the server.

Time out

INPUT; BINARY(4)

The amount of time to wait for a client response. The value specified is in units of seconds. Valid values are 1 through 100.

Note: This is not a time-out value for this API. It is a time-out for each SNMP call that this API makes to the client.

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
CPFA835 E	Time-out parameter not correct.
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5E87 E	Information type not valid.
CPF5E88 E	Parameter length not valid.
CPF5E89 E	Client handle does not exist.
CPF5E8E E	Unable to contact client.
CPF5E94 E	Client information not refreshed.
CPF5E95 E	Internal processing error.
CPF9845 E	Error occurred while opening file &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Remove Client (QZCARMVC, QzcaRemoveClient) API

Required Parameter Group:

1	Client handle	Input	Char(12)
2	Error code	I/O	Char(*)

Service Program: QZCACL T

Default Public Authority: *EXCLUDE

Threadsafe: No

The Remove Client (OPM, QZCARMVC; ILE, QzcaRemoveClient) API allows applications to make clients not known to the server. All hardware and software information associated with the client is also removed.

Authorities and Locks

API Public Authority

*EXCLUDE

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

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
CPF24B4 E	Severe error while addressing parameter list.

CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5E88 E	Parameter length not valid.
CPF5E89 E	Client handle does not exist.
CPF5E95 E	Internal processing error.
CPF9845 E	Error occurred while opening file &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Update Client Information (QZCAUPDC, QzcaUpdateClientInfo) API

Required Parameter Group:

1	Client handle	Input	Char(12)
2	Client information list	Input	Char(*)
3	Length of client information list	Input	Binary(4)
4	Error code	I/O	Char(*)

Service Program: QZCACL T

Default Public Authority: *EXCLUDE

Threadsafe: No

The Update Client Information (OPM, QZCAUPDC; ILE, QzcaUpdateClientInfo) API allows applications to update some miscellaneous information stored on the system about a client.

Authorities and Locks

API Public Authority

*EXCLUDE

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Client information list

INPUT; CHAR(*)

The information that should be updated at the server. For the format of this structure, see [Client Information List](#).

Length of client information list

INPUT; BINARY(4)

The length of the client information list area in bytes. Valid values are 16 through 1016.

Error code

I/O; CHAR(*)

The structure in which to return error information. For the format of the structure, see [Error Code Parameter](#).

Client Information List

The client information list will have the following form:

Name	Type	Description
Number of tokens	BINARY(4)	The number of tokens, which is the same number of offsets, in the list. Valid values are 1 through 9.
Array of offsets to tokens	Array of BINARY(4)	Offset to token
...
Token	BINARY(4)	The field to be changed
Length of value	BINARY(4)	The length of the value
Value	CHAR(*)	The new value for the field to be changed
...

Notes:

1. Offsets are calculated from the beginning of the client information list.
2. It is the responsibility of the calling application to ensure that all binary fields are aligned on a 4-byte boundary.

Business Location (Token=1): The new value for the business location field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of business location
8	8	CHAR(*)	Business location

Community Name (Token=2): The new value for the community name field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of community name
8	8	CHAR(*)	Community name

Description (Token=3): The new value for the description field.

Offset		

Dec	Hex	Type	Field
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of description
8	8	CHAR(*)	Description

Internet Address (Token=4): The new value for the internet address field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of internet address
8	8	CHAR(*)	Internet address

Office (Token=5): The new value for the office field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of office
8	8	CHAR(*)	Office

Owner (Token=6): The new value for the owner field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of owner
8	8	CHAR(*)	Owner

Owner Telephone (Token=7): The new value for the owner telephone field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of owner telephone
8	8	CHAR(*)	Owner telephone

Client Access User Profile (Token=8): The new value for the Client Access user profile field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of Client Access user profile
8	8	CHAR(*)	Client Access user profile

Client Access Status (Token=9): The new value for the Client Access status field.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Token
4	4	BINARY(4)	Length of Client Access status
8	8	CHAR(*)	Client Access status

Field Descriptions

Business location. The physical location of this client.

Client access status. The status of the Client Access client.

The valid special values follow:

**ACTIVE* Client Access has identified this client as active.

**INACTIVE* Client Access has identified this client as inactive.

Client Access user profile. An iSeries user profile.

Community name. A relationship between a Simple Network Management Protocol (SNMP) agent and a set of SNMP managers that are used for authentication, access control, and proxy.

Description. A user-defined description for this client.

Internet address. The internet address is specified in the form nnn.nnn.nnn.nnn. Values for nnn range from 0 through 255.

Length of business location. Values are from 0 through 255.

Length of community name. Values are 0 through 255.

Length of Client Access status. Values are from 7 through 10.

Length of Client Access user profile. Values are from 0 through 10.

Length of description. Values are from 0 through 255.

Length of internet address. Values are from 7 through 15.

Length of office. Values are from 0 through 32.

Length of owner. Values are from 0 through 32.

Length of owner telephone. Values are from 0 through 32.

Office. The office of the owner for this client.

Owner. The owner for this client.

Owner telephone. The telephone number of the owner for this client.

Token. An integer value that defines the token you are working with.

Error Messages

Message ID	Error Message Text
CPF24B4 E	Severe error while addressing parameter list.
CPF3C1E E	Required parameter &1 omitted.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CF1 E	Error code parameter not valid.
CPF5E88 E	Parameter length not valid.
CPF5E89 E	Client handle does not exist.
CPF5E95 E	Internal processing error.
CPF9845 E	Error occurred while opening file &1.
CPF9846 E	Error while processing file &1 in library &2.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Add Client Exit Program (XZCADDCC)

Required Parameter Group:

1	Client handle	Input	Char(12)
---	---------------	-------	----------

Exit Point Name: QIBM_QZCA_ADDC

Exit Point Format Name: ZCAA0100

The Add Client exit program allows applications to be notified when clients are made known to the server. The client application is called with the client handle parameter.

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Exit program introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Lotus Notes Database Exit Program

Required Parameter Group:

1	Change table name	Input	Char(10)
2	Notes database name	Input	Char(112)
3	DB2 collection or library name	Input	Char(10)
4	DB2 table name	Input	Char(128)

Note: To use this exit program, you need the IBM OS/400 Lotus Notes Enhanced Integration feature. This feature is not available on releases past V4R3.

The Lotus Notes Database exit program provides the capability for the user to process an SQL table of all changes (updates, insertions, and deletions) that have occurred in the Notes database that is being shadowed from DB2 UDB for iSeries.

The Lotus Notes Database exit program refers to the OS/400 program that the user specifies from the Notes interface to the Register Notes to DB2 UDB for OS/400 Exit Program option.

The input parameters are padded with blanks to the maximum length of each parameter. They are not null-terminated. Therefore, if you need to compare or use the values of these parameters in a C-language program, your exit program may need to remove any trailing blanks before use.

An SQL table is built in the library of the DB2 collection or library where the base table of the import and shadow request resides. This change table contains a row for each change that occurred in the Notes database as a result of a Notes client operation. The row contains the operation, column values, and key values of the operations.

The exit program is submitted as a batch job. Any errors that the exit program generates are sent to the job log of the batch job.

For more information, see the [Globalization](#) topic.

Required Parameter Group

Change table name

INPUT; CHAR(10)

The name of the change table in the base table collection or library (as specified on the original shadow request) that contains the summary of changes to the Notes database.

Note: Only changes made by a Notes client will be placed in this table.

Notes database name

INPUT; CHAR(112)

The name of the Notes database in which the changes have occurred.

DB2 collection or library name

INPUT; CHAR(10)

The name of the DB2 collection or library where the change table and base table reside.

DB2 table name

INPUT; CHAR(128)

The name of the DB2 base table that is shadowed to the Notes server. This is the name of the table specified on the original import or shadow SQL statement.

Format of Change Table

The following table shows the format of the change table. For a description of the fields in this format, see [Field Descriptions](#).

Offset		Type	Field
Dec	Hex		
0	0	CHAR(1)	OPERATION
1	1	CHAR(26)	TIMESTAMP
27	1B	CHAR(256)	UPDATEDBY
283	11B	CHAR(256)	FORM
539	21B	BINARY(4)	DOCID
...			

Field Descriptions

DOCID. The Notes ID of the document where the operation took place in the Notes database.

FORM. The Notes form name that was used to perform the operation on the Notes database.

OPERATION. The operation flag that indicates whether the imported DB2 UDB for iSeries database data row is inserted, updated, or deleted. Possible values follow:

I Insert

D Delete

U Update

TIMESTAMP. The timestamp in the Notes document of when the operation occurred. The 26-character format is YYYY-MM-DD-HH.MM.SS.ssssss, where:

YYYY Year

MM Month

DD Day
HH Hour
MM Minute
SS Second
sssss Microsecond

UPDATEDBY. The Notes user who performed the operation on the Notes database.

... The values and offsets from here on are run-time dependent. The columns included in the change table will be the same as the order and attributes of the columns that were selected in the original SQL statement that was used for the import and shadow request. If no key columns were specified on the original import statement, key columns are placed first in the table. In addition, each key column is preceded by a column that contains the old key value.

Exit program introduced: V3R7

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Refresh Client Information Exit Program

Required Parameter Group:

1	Client handle	Input	Char(12)
---	---------------	-------	----------

Exit Point Name: QIBM_QZCA_REFC

Exit Point Format Name: ZCAF0100

The Refresh Client Information exit program allows applications to be notified when information about clients has been refreshed through the Refresh Client Information API from the server. The client application is called with the client handle parameter.

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Exit program introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Remove Client Exit Program

Required Parameter Group:

1	Client handle	Input	Char(12)
---	---------------	-------	----------

Exit Point Name: QIBM_QZCA_RMVC

Exit Point Format Name: ZCAR0100

The Remove Client Exit Program allows applications to be notified when clients are removed from the server. The client application is called with the client handle parameter.

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Exit program introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)

Update Client Information Exit Program

Required Parameter Group:

1	Client handle	Input	Char(12)
---	---------------	-------	----------

Exit Point Name: QIBM_QZCA_UPDC

Exit Point Format Name: ZCAU0100

The Update Client Information exit program allows applications to be notified when information about clients has been updated through the Update Client Information API from the server. The client application is called with the client handle parameter.

Required Parameter Group

Client handle

INPUT; CHAR(12)

A unique key to identify this managed node. This parameter was returned in the client handle assigned parameter on the Add Client API.

Exit program introduced: V3R6

[Top](#) | [Client Management Support APIs](#) | [APIs by category](#)