



System i
Programming
Network Security APIs

Version 6 Release 1





System i
Programming
Network Security APIs

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in "Notices," on page 11.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-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.

© Copyright International Business Machines Corporation 1998, 2008. All rights reserved.

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

Contents

Network Security APIs	1
Server Authentication Entry APIs	1
APIs	1
Add Server Authentication Entry (QsyAddServerEntry) API	1
Authorities and Locks	2
Required Parameter Group	2
Error Messages	2
Change Server Authentication Entry (QsyChangeServerEntry) API	3
Authorities and Locks	3
Required Parameter Group	3
Error Messages	4
Remove Server Authentication Entry (QsyRemoveServerEntry) API	4
Authorities and Locks	5

Parameters	5
Error Messages	5
Retrieve Server Authentication Entries (QSYRTVSE, QsyRetrieveServerEntries) API	6
Authorities and Locks	6
Required Parameter Group	6
Receiver Variable Description	8
Field Descriptions	8
Error Messages	9

Appendix. Notices	11
Programming interface information	12
Trademarks	13
Terms and conditions	14

Network Security APIs

The network security APIs include:

- “Server Authentication Entry APIs”

Security APIs | APIs by category

Server Authentication Entry APIs

The Server Authentication Entry APIs can be used by a client requesting connection to a server.

The server authentication entry APIs are:

- “Add Server Authentication Entry (QsyAddServerEntry) API” (QsyAddServerEntry) adds server authentication information for use by application requesters in connecting to application servers.
- “Change Server Authentication Entry (QsyChangeServerEntry) API” on page 3 (QsyChangeServerEntry) changes server authentication information for use by application requesters in connecting to application servers.
- “Remove Server Authentication Entry (QsyRemoveServerEntry) API” on page 4 (QsyRemoveServerEntry) removes server authentication information for use by application requesters in connecting to application servers.
- “Retrieve Server Authentication Entries (QSYRTVSE, QsyRetrieveServerEntries) API” on page 6 (QSYRTVSE, QsyRetrieveServerEntries) returns a list of server authentication entries for a user profile.

Security APIs | APIs by category

APIs

These are the APIs for this category.

Add Server Authentication Entry (QsyAddServerEntry) API

Syntax for QsyAddServerEntry:

```
#include <qsysvrfn.h>

void QsyAddServerEntry
(char          User_profile[10],
char          *Server_name,
int           Length_of_server_name,
char          *User_ID,
int           Length_of_user_ID,
char          *Password,
int           Length_of_password,
void          *Error_code);
```

Service Program: QSYSVRFN
Default Public Authority: *USE
Threadsafe: No

The Add Server Authentication Entry (QsyAddServerEntry) API adds server authentication information for use by application requesters in connecting to application servers.

When adding a server authentication entry for a Distributed Relational Database Architecture™ (DRDA®) application that uses TCP/IP, the server name must be entered in upper case.

Authorities and Locks

If the user profile parameter is not *CURRENT or the user profile currently running, then the user profile that calls this API must have *SECADM special authority and *OBJMGT and *USE authorities to the user profile.

Required Parameter Group

User profile

INPUT; CHAR(10)

The user profile for which the server authentication entry will be added. The special value *CURRENT may be specified to add an entry for the user profile that calls this API.

Server name

INPUT; CHAR(*)

The name of the application server.

Length of server name

INPUT; BINARY(4)

The length of the server name that is specified in the server name parameter. The length of the server name must be a value from 1 to 200.

User ID

INPUT; CHAR(*)

The user name for which requests will be made to the implementation server.

Length of user ID

INPUT; BINARY(4)

The length of the user ID that is specified in the user ID parameter. The length of the user ID must be a value from 0 to 1000. If the length is 0, the user ID will be the same as the name that is specified in the user profile parameter.

Password

INPUT; CHAR(*)

The password to be used to authenticate the user when the client attempts to connect to the server.

Length of password

INPUT; BINARY(4)

The length of the password that is specified in the password parameter. The length of the password must be a value from 0 to 696. If the length is 0, then no password is supplied on the connection request. If the retain server security data (QRETSVRSEC) system value is set to 0 (do not retain data), then the length of the password is assumed to be 0.

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
CPF2204 E	User profile &1 not found.

Message ID	Error Message Text
CPF2213 E	Not able to allocate user profile &1.
CPF2222 E	Storage limit is greater than specified for user profile &1.
CPF224F E	Server authentication entry already exists.
CPF225F E	Not all information stored.
CPF226C E	Not authorized to perform function.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

[Top](#) | [Security APIs](#) | [APIs by category](#)

Change Server Authentication Entry (QsyChangeServerEntry) API

Syntax for QsyChangeServerEntry:

```
#include <qsysvrfn.h>

void QsyChangeServerEntry
(char          User_profile[10],
char          *Server_name,
int           Length_of_server_name,
char          *User_ID,
int           Length_of_user_ID,
char          *Password,
int           Length_of_password,
void          *Error_code);
```

Service Program: QSYSVRFN
Default Public Authority: *USE
Threadsafe: No

The Change Server Authentication Entry (QsyChangeServerEntry) API changes the server authentication information for use by application requesters in connecting to application servers.

Authorities and Locks

If the user profile parameter is not *CURRENT or the user profile currently running, then the user profile that calls this API must have *SECADM special authority and *OBJMGT and *USE authorities to the user profile.

Required Parameter Group

User profile

INPUT; CHAR(10)

The user profile for which the server authentication entry will be changed. The special value *CURRENT may be specified to change an entry for the user profile that calls this API.

Server name

INPUT; CHAR(*)

The name of the application server.

Length of server name

INPUT; BINARY(4)

The length of the server name that is specified in the server name parameter. The length of the server name must be a value from 1 to 200.

User ID

INPUT; CHAR(*)

The user name for which requests will be made to the implementation server.

Length of user ID

INPUT; BINARY(4)

The length of the user ID that is specified in the user ID parameter. The length of the user ID must be a value from -1 to 1000. If -1 is specified, the user ID value is not changed. If 0 is specified, the user ID will be the same as the name that is specified in the user profile parameter.

Password

INPUT; CHAR(*)

The password to be used to authenticate the user when the client attempts to connect to the server.

Length of password

INPUT; BINARY(4)

The length of the password that is specified in the password parameter. The length of the password must be a value from -1 to 696. If -1 is specified, the password value does not change. If 0 is specified, then no password is supplied on the connection request. If the retain server security data (QRETSVRSEC) system value is set to 0 (do not retain data), then the length of the password is assumed to be 0.

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
CPF2204 E	User profile &1 not found.
CPF2213 E	Not able to allocate user profile &1.
CPF225E E	Server authentication entry does not exist.
CPF225F E	Not all information stored.
CPF226C E	Not authorized to perform function.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

[Top](#) | [Security APIs](#) | [APIs by category](#)

Remove Server Authentication Entry (QsyRemoveServerEntry) API

Syntax for QsyRemoveServerEntry:

```
#include <qsysvrfn.h>

void QsyRemoveServerEntry(
```

```

char    User_profile[10],
char    *Server_name,
int     Length_of_server_name,
void    *Error_code);

```

Service Program: QSYSVRFN
 Default Public Authority: *USE
 Threadsafte: No

The Remove Server Authentication Entry (QsyRemoveServerEntry) API removes server authentication information for use by application requesters in connecting to application servers.

Authorities and Locks

If the user profile parameter is not *CURRENT or the user profile currently running, then the user profile that calls this API must have *SECADM special authority and *OBJMGT and *USE authorities to the user profile.

Parameters

User profile

INPUT; CHAR(10)

The user profile for which the server authentication entry will be removed. The special value *CURRENT may be specified to remove an entry for the user profile that calls this API.

Server name

INPUT; CHAR(*)

The name of the application server. The special value *ALL may be specified to indicate that all server authentication entries for the user profile that is specified in the user profile parameter are to be removed.

Length of server name

INPUT; BINARY(4)

The length of the server name that is specified in the server name parameter. The length of the server name must be a value from 1 to 200.

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
CPF2204 E	User profile &1 not found.
CPF2213 E	Not able to allocate user profile &1.
CPF225E E	Server authentication entry does not exist.
CPF226C E	Not authorized to perform function.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Retrieve Server Authentication Entries (QSYRTVSE, QsyRetrieveServerEntries) API

Required Parameter Group for QSYRTVSE:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Return records feedback information	Output	Char(12)
4	Format name	Input	Char(8)
5	Starting server name	Input	Char(*)
6	Length of starting server name	Input	Binary(4)
7	Starting server option	Input	Char(1)
8	User profile	Input	Char(10)
9	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: No

Syntax for QsyRetrieveServerEntries:

```
#include <qsysvrfn.h>
```

```
void QsyRetrieveServerEntries  
(void                *Receiver_variable,  
 int                 Length_of_receiver_variable,  
 Qsy_RSvRE_Feedback_Info_T *Return_records_feedback_information,  
 char                *Format_name,  
 char                *Starting_server_name,  
 int                 Length_of_starting_server_name,  
 char                Starting_server_option,  
 char                User_profile[10],  
 void                *Error_code);
```

Service Program: QSYSVRFN
Default Public Authority: *USE
Threadsafe: No

The Retrieve Server Authentication Entries (OPM, QSYRTVSE; ILE, QsyRetrieveServerEntries) API returns a list of server authentication entries for a user profile.

Authorities and Locks

User Profile Authority
*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 that the area can hold.

Length of receiver variable

INPUT; BINARY(4)

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

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

Returned records feedback information

OUTPUT; CHAR(12)

Information about the entries that are returned in the receiver variable.

See "Format of Returned Records Feedback Information" on page 8 for details.

Format name

INPUT; CHAR(8)

The name of the format that is used to retrieve server authentication entries for the user profile.

You can specify this format:

SVRE0100 For a detailed description of this format, see "SVRE0100 Format" on page 8.

Starting server name

INPUT; CHAR(*)

The server name at which to start listing server authentication entries. The server authentication entries are listed in hexadecimal sort sequence by server name.

Possible values follow:

- *FIRST* Server authentication entries are returned starting with the server that has the smallest hexadecimal value.
- server name* If an exact match for the starting server name is found, the starting server option parameter indicates whether that server authentication entry is returned.
If an exact match for the starting server name is not found, the listing begins with the first existing server authentication entry for the server name whose hexadecimal value would follow the hexadecimal value of the specified starting server name.

Length of starting server name

INPUT; BINARY(4)

The length of the starting server name. The length of the starting server name may be from 1 to 200.

Starting server option

INPUT; CHAR(1)

This parameter indicates whether the starting server authentication entry is returned when an exact match for the starting server name is found.

Possible values follow:

- 0* Server authentication entries for server names whose hexadecimal value is greater than the hexadecimal value for the starting server name are returned.
- 1* Server authentication entries for server names whose hexadecimal value is equal to or greater than the hexadecimal value for the starting server name are returned.

User profile

INPUT; CHAR(10)

The name of the user profile for which the server authentication entries are returned.

Error code

I/O; CHAR(*)

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

Receiver Variable Description

The following tables describe the order and format of the data returned in the receiver variable. For detailed descriptions of the fields in the tables, see “Field Descriptions.”

SVRE0100 Format

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of entry
4	4	BINARY(4)	Length of server name
8	8	BINARY(4)	CCSID of server name
12	C	CHAR(200)	Server name
212	D4	BINARY(4)	Displacement to user ID
216	D8	BINARY(4)	Length of user ID
220	DC	BINARY(4)	CCSID of user ID
224	E0	CHAR(1)	Password stored indicator
		CHAR(*)	User ID

Format of Returned Records Feedback Information

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Number of server authentication entries

Field Descriptions

Bytes available. The number of bytes of data available to be returned to the user in the receiver variable. If all data is returned, bytes available is the same as the number of bytes returned. If the receiver variable was not large enough to contain all of the data, this value is estimated based on the total number of server authentication entries for the user profile and the format specified.

Bytes returned. The number of bytes of data returned to the user in the receiver variable. This is the lesser of the number of bytes available to be returned or the length of the receiver variable.

CCSID of server name. The CCSID of the server name. This will be the default job CCSID of the job that added the server authentication entry.

CCSID of user ID. The CCSID of the user ID. This will be the default job CCSID of the job that last changed the user ID field in the server authentication entry.

Displacement to user ID. The displacement in the entry to the start of the user ID.

Format name. The name of the format that is used to return server authentication entries for a user profile.

Length of entry. The length (in bytes) of the current entry. This length can be used to access the next entry.

Length of server name ID. The length (in bytes) of the server name.

Length of user ID. The length (in bytes) of the user ID.

Number of server authentication entries. The number of complete entries returned in the list of server authentication entries. A value of zero is returned if the list is empty.

Password specified. Indicates whether the server authentication entry has a password associated with it.

Possible values follow:

0 The server authentication entry does not have a password associated with it.

1 The server authentication entry does have a password associated with it.

Server name. The name of the server that the entry is for.

User ID. The user ID that is used on requests to the server.

User profile. The name of the user profile for which the list of server authentication entries is returned.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF2204 E	User profile &1 not found.
CPF2213 E	Not able to allocate user profile &1.
CPF2217 E	Not authorized to user profile &1.
CPF2222 E	Storage limit is greater than specified for user profile &1.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

[Top](#) | [Security APIs](#) | [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 document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This API descriptions 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 Presentation
Advanced Peer-to-Peer Networking
AFP
AIX
AnyNet
AS/400
BCOCA
C/400
COBOL/400
Common User Access
CUA
DB2
DB2 Universal Database
Distributed Relational Database Architecture
Domino
DPI
DRDA
Enterprise Storage Server
eServer
FlashCopy
GDDM
i5/OS
IBM
IBM (logo)
InfoColor
Infoprint
Integrated Language Environment
Intelligent Printer Data Stream
IPDS
Lotus
Lotus Notes
MO:DCA
MVS
Net.Data
NetServer
Notes
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/400
PartnerWorld
POWER5+
PowerPC
Print Services Facility
PrintManager
PROFS
RISC System/6000
RPG/400
RS/6000

SAA
SecureWay
SOM
System i
System i5
System Object Model
System/36
System/38
System/390
TotalStorage
VisualAge
WebSphere
xSeries
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER

EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Printed in USA