System i

Programming
Digital Certificate Management APIs

Version 6 Release 1

System i

Programming
Digital Certificate Management APIs

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in “Notices,” on page 85.

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.

Contents

Digital Certificate Management APIs

The digital certificate management APIs enable X.509 type certificates to be associated with a user profile.The APIs add, remove, list, and find certificates that are associated with user profiles.

This section also includes APIs for registering applications that use certificates. Applications that need to use certificates will make themselves known by registering themselves. As part of that registration, applications will identify an exit program that is to be called:

- whenever a certificate is assigned to the application or if the certificate assignment changes.
- whenever a Certificate Authority (CA) is added to or removed from the trust list for the application.
- whenever the information about the application is being changed.
- whenever the application is being deregistered.

The application is, therefore, not responsible for providing a user interface for certificate management. When the application starts, it can retrieve the name and location of the certificate assigned to the application and use it for initiating a Secure Sockets Layer (SSL) session or some other operation that requires a certificate.

The digital certificate management APIs are:

- "Add User Certificate (QSYADDUC, QsyAddUserCertificate) API" on page 2 (QSYADDUC, QsyAddUserCertificate) associates a certificate with an i5/OS user profile.
- "Add Validation List Certificate (QSYADDVC, QsyAddVldlCertificate) API" on page 4 (QSYADDVC,QsyAddVldlCertificate) adds a certificate to a validation list.
- "Check Validation List Certificate (QSYCHKVC,QsyCheckVldlCertificate) AP" on page 6 (QSYCHKVC, QsyCheckVldlCertificate) determines whether a certificate is in a validation list.
- "Deregister Application for Certificate Use (QSYDRGAP, QsyDeregisterAppForCertUse) API" on page 8 (QSYDRGAP, QsyDeregisterAppForCertUse) removes an application and all associated certificate information from the registration facility.
- "Export Certificate Store (QYKMEXPK, QykmExportKeyStore)" on page 9 (QYKMEXPK, QykmExportKeyStore)) exports a certificate store to a PKCS 12 version 3 standard file.
- "Find Certificate User (QSYFNDCU,QsyFindCertificateUser) API" on page 12 (QSYFNDCU, QsyFindCertificateUser) finds the user that is associated with a certificate.
- "Generate and Sign User Certificate Request (QYCUGSUC) API" on page 14 (QYCUGSUC) generates a user certificate request and then signs the certificate request using the local Certificate Authority (CA).
- "Get Default Key Item (QYKMGDKI, QykmGetDefaultKeyItem)" on page 16 (QYKMGDKI, QykmGetDefaultKeyItem) Allows you to retrieve the label of the default certificate in a certificate store.
- "Import Certificate Store (QYKMIMPK, QykmImportKeyStore)" on page 18 (QYKMIMPK, QykmImportKeyStore)) imports a certificate store from a PKCS 12 version 3 standard file.
- "List User Certificates (QSYLSTUC, QsyListUserCertificates) API" on page 21 (QSYLSTUC, QsyListUserCertificates) lists the certificates in the user profile.
- "List Validation List Certificates (QSYLSTVC,QsyListVldlCertificates) API" on page 31 (QSYLSTVC, QsyListVldlCertificates) lists the certificates in the validation list.
- "Open List of User Certificates (QSYOLUC) API" on page 36 (QSYOLUC) provides a list of user certificates associated with a user.
- "Parse Certificate (QSYPARSC, QsyParseCertificate) API" on page 39 (QSYPARSC, QsyParseCertificate) parses a certificate and puts the results in the caller's storage.
- "Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API" on page 46 (QSYRGAP, QsyRegisterAppForCertUse) registers an application with the registration facility.

- "Remove User Certificate (QSYRMVUC, QsyRemoveUserCertificate) API" on page 52 (QSYRMVUC, QsyRemoveUserCertificate) removes a certificate from an i5/OS user profile.
- "Remove Validation List Certificate (QSYRMVVC, QsyRemoveVldlCertificate) API" on page 54 (QSYRMVVC, QsyRemoveVldlCertificate) removes a certificate from a validation list.
- » "Retrieve Certificate Information (QYCURTVCI, QycuRetrieveCertificateInfo) API" on page 56 (QYCURTVCI, QycuRetrieveCertificateInfo) retrieves information from server or CA certificates. «
- "QsyRetrieveDigitalIDConfig()—Retrieve Digital ID Configuration Information" on page 67 (QsyRetrieveDigitalIDConfig()) retrieves digital ID configuration information.
- "QsySetDigitalIDConfig()—Set Digital ID Configuration Information" on page 70 (QsySetDigitalIDConfig()) sets digital ID configuration information.
- "Sign User Certificate Request (QYCUSUC) API" on page 73 (QYCUSUC) signs a user certificate request using the local Certificate Authority (CA).

Note: All of these APIs, except Register and Deregister Application for Certificate Use, require that Digital Certificate Manager, option 34 of the i5/OS® licensed program (5761-SS1), be installed.

APIs

These are the APIs for this category.

Add User Certificate (QSYADDUC, QsyAddUserCertificate) API

Required Parameter Group for QSYADDUC:

1	User profile	Input	Char(10)
2	Certificate	Input	Char(*)
3	Type	Input	Binary(4)
4	Length of certificate	Input	Binary(4)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyAddUserCertificate:

```
#include <qsydigid.h>

void QsyAddUserCertificate
       (char           *User_profile,
        char           *Certificate,
        int             Type,
        int             Length_of_certificate,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Add User Certificate (OPM, QSYADDUC; ILE, QsyAddUserCertificate) API associates a certificate with an i5/OS® user profile.

A common scenario is that only one certificate is associated with an i5/OS user profile at any given time, but more than one certificate may be associated with the same i5/OS user profile if each certificate is

unique. A reason for having more than one certificate associated with an i5/OS user profile may be that the first certificate is about to expire. The same certificate is not allowed to be associated with more than one i5/OS user profile.

Because certificates vary in length, the actual number of certificates that can be listed using the List User Certificates API will also vary. Depending on the length of each of the certificates, no more than a few hundred certificates should be added to an i5/OS user profile or incomplete results may be returned when attempting to use the List User Certificates API to list certificates that are associated with the i5/OS user profile.

Authorities and Locks

User Profile Authority
If the user profile specified is not the user profile that is currently running, then *SECADM special authority and *USE and *OBJMGT authorities to the user profile are required.

Required Parameter Group

User profile
INPUT; CHAR(10)

The name of the user profile that will hold the certificate.

The following is also a valid selection for the user profile:

**CURRENT* The user profile that is currently running.

Certificate
INPUT; CHAR(*)

The entire certificate in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) format. This is not a text string. This certificate is associated with the user profile.

Type INPUT; BINARY(4)

The type or format of the certificate.

The possible types are:

1 Entire X.509 public key certificate in ASN.1 DER encoding.
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate.

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
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF2204 E	User profile &1 not found.

Message ID	Error Message Text
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.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF227C E	Certificate association already exists.
CPF3BFF E	Required option &1 is not available.
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.
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.
CPF4AB9 E	User certificate function not successful.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Add Validation List Certificate (QSYADDVC, QsyAddVldlCertificate) API

Required Parameter Group for QSYADDVC:

1	Validation list path name	Input	Char(*)
2	Length of path	Input	Binary(4)
3	Certificate	Input	Char(*)
4	Type	Input	Binary(4)
5	Length of certificate	Input	Binary(4)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyAddVldlCertificate:

```
#include <qsydigid.h>

void QsyAddVldlCertificate
       (char    *Validation_list_path_name,
        int      Length_of_path,
        char    *Certificate
        int      Type,
        int      Length_of_certificate,
        void    *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Add Validation List Certificate (OPM, QSYADDVC; ILE, QsyAddVldlCertificate) API adds a certificate to a validation list.

It is likely that many certificates will be added to a validation list. Each certificate that is added to a validation list must be unique in that validation list. The same certificate can be added to more than one validation list.

Authorities and Locks

Validation List Authority
*USE and *ADD

Validation List Library Authority
*EXECUTE

Required Parameter Group

Validation list path name
INPUT; CHAR(*)

The fully qualified path name of the validation list.
Example value:
/QSYS.LIB/SMITH.LIB/EXAMPLE.VLDL

Length of path
INPUT; BINARY(4)

The length of the validation list path.

Certificate
INPUT; CHAR(*)

The entire X.509 certificate encoded in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) format. This is not a text string.

Type INPUT; BINARY(4)

The type of the certificate.

The possible types are:

1 Entire X.509 public key certificate in ASN.1 DER encoding.
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate.

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
CPFA0AA E	Error occurred while attempting to obtain space.
CPFA09C E	Not authorized to object.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF227C E	Certificate association already exists.
CPF3BFF E	Required option &1 is not available.
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.

Message ID	Error Message Text
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Check Validation List Certificate (QSYCHKVC,QsyCheckVldlCertificate) AP

Required Parameter Group for QSYCHKVC:

1	Validation list path name	Input	Char(*)
2	Length of path	Input	Binary(4)
3	Certificate	Input	Char(*)
4	Type	Input	Binary(4)
5	Length of certificate	Input	Binary(4)
6	Return code	Output	Binary(4)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyCheckVldlCertificate:

```
#include <qsydigid.h>

void QsyCheckVldlCertificate
       (char           *Validation_list_path_name,
        int             Length_of_path,
        char           *Certificate,
        int             Type,
        int             Length_of_certificate,
        int            *Return_code,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Check Validation List Certificate (OPM, QSYCHKVC; ILE, QsyCheckVldlCertificate) API determines whether a certificate is in a validation list.

Authorities and Locks

Validation List Authority
*USE

Validation List Library Authority
*EXECUTE

Required Parameter Group

Validation list path name
INPUT; CHAR(*)

The fully qualified path name of the validation list.

Length of path
INPUT; BINARY(4)

The length of the validation list path.

Certificate
INPUT; CHAR(*)

The certificate or the handle of the certificate to be checked. This is not a text string.

Type INPUT; BINARY(4)

The type of the certificate.

The possible types are:

1 Entire X.509 public key certificate in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) encoding.
2 Certificate handle of X.509 certificate
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate that was provided. The type parameter indicates what this length refers to.

Return code
OUTPUT; BINARY(4)

The return code that indicates the result of the check.

The possible types are:

1 Certificate was found in the validation list.
0 Certificate was not found in the validation list.

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
CPFA0AA E	Error occurred while attempting to obtain space.

Message ID	Error Message Text
CPFA09C E	Not authorized to object.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF3BFF E	Required option &1 is not available.
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.
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Deregister Application for Certificate Use (QSYDRGAP, QsyDeregisterAppForCertUse) API

Required Parameter Group for QSYDRGAP:

1	Application ID	Input	Char(*)
2	Length of application ID	Input	Binary(4)
3	Error code	I/O	Char(*)

Default Public Authority: *EXCLUDE
Threadsafe: Yes

Syntax for QsyDeregisterAppForCertUse:

```
#include <qsyrgap1.h>

void QsyDeregisterAppForCertUse
       (char           *Application_ID,
        int            *Length_of_application_ID,
        void           *Error_code);
```

Service Program: QSYRGAP1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Deregister Application for Certificate Use (OPM, QSYDRGAP; ILE, QsyDeregisterAppForCertUse) API removes an application and all associated certificate information from the registration facility. When an object signing application is deregistered, the corresponding function with the same ID also will be deregistered (see Deregister Function (QSYDRGFN, QsyDeregisterFunction) API). The corresponding function was registered when the object signing application was registered.

Authorities and Locks

API Public Authority
*EXCLUDE

Registration Lock
*EXCL

Required Parameter Group

Application ID
INPUT; CHAR(*)

The ID for the application being removed.

The following can be specified for the application ID:

*generic**	All applications that have IDs beginning with the generic string.
application ID	Specific application ID.

Length of application ID
INPUT; BINARY(4)

The length of the specified application ID. The length must be a value from 1 to 100.

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
CPF220E E	Application &1 not registered.
CPF3C90 E	Literal value cannot be changed.
CPF3CD9 E	Requested function cannot be performed at this time.
CPF3CDA E	Registration facility repository not available for use.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9810 E	Library &1 not found.
CPF9811 E	Program &1 in library &2 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R4

Top | Security APIs | APIs by category

Export Certificate Store (QYKMEXPK, QykmExportKeyStore)

Required Parameter Group:

1	Certificate store path and file Name	Input	Char(*)
2	Length of certificate store path and file Name	Input	Binary(4)
3	Format of certificate store path and file Name	Input	Char(8)
4	Certificate store password	Input	Char(*)
5	Length of certificate store password	Input	Binary(4)
6	CCSID of certificate store password	Input	Binary(4)

7	Export path and file name	Input	Char(*)
8	Length of export path and file name	Input	Binary(4)
9	Format of export path and file name	Input	Char(8)
10	Version of export file	Input	Char(10)
11	Export file password	Input	Char(*)
12	Length of export file password	Input	Binary(4)
13	CCSID of export file password	Input	Binary(4)
14	Error code	I/O	Char(*)

Service Program Name: QYKMSYNC
Default Public Authority: *USE
Threadsafe: No

The Export Certificate Store API (OPM, QYKMEXPK; ILE, QykmEportKeyStore) allows a user to export an entire certificate store to a PKCS12 version 3 standard file. This allows for the export of private keys as well as record labels. **Note:** Option 34, Digital Certificate Manager, must be installed in order to use this API.

Authorities and Locks

Authority Required

To use this API, option 34 must be installed. You must also know the password of the certificate store if you want to export private keys. Additionally, you must provide a password for the export file which gets created as a result of calling this API.

For the file objects:

- *R authority to the certificate store.
- *RW authority to the export file.

For the directories:

- *WX authority to the directory containing the export file.
- *X authority to each directory in the paths to both the certificate store and export files.

Also, see the open()—Open File API for the authority needed to the certificate store. The export file must not exist prior to calling this API. The export file gets created as a result of calling this API.

Locks Object will be locked shared read.

Required Parameter Group

Certificate store path and file name
INPUT; Char(*)

The path name of the certificate store (kdb) you want to export. This path and file name may be absolute (i.e., entire path name) or relative to the current directory. If you are using format OBJN0100 (see below), this parameter is assumed to be represented in the coded character set identifier (CCSID) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Length of certificate store path and file name
INPUT; Binary(4)

The length of the certificate store path and file name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of certificate store path and file name
INPUT; CHAR(8)

The format of the certificate store path and file name parameter.

OBJN0100 The certificate store path and file name is a simple path name.
OBJN0200 The certificate path and file name is an LG-type path name.

Certificate store password
INPUT; CHAR(*)

The password of the certificate store whose certificates will be exported to the given export file. If the password parameter is null, private keys will not be exported.

Length of certificate store password
INPUT; Binary(4)

The length of the password of the certificate store whose certificates will be exported to the given export file. If the length of the password is 0, private keys will not be exported.

CCSID of certificate store password
INPUT; Binary(4)

This parameter is the CCSID of the certificate store password. If the value is 0, the default CCSID of the job will be used.

Export path and file name
INPUT; CHAR(*)

The path (including the name) of the export file into which all of the certificates in the certificate store will be exported in the format indicated by the version of the export file parameter. This path and file name may be absolute (i.e., entire path name) or relative to the current directory. If you are using format OBJN0100 (see below), this parameter is assumed to be represented in the coded character set identifier (CCSID) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Length of export path and file name
INPUT; Binary(4)

The length of the export path and file name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of export path and file name
INPUT; CHAR(8)

The format of the export path and file name parameter.

OBJN0100 The export path and file name is a simple path name.
OBJN0200 The export path and file name is an LG-type path name.

Version of export file
INPUT; Char(10)

Currently, the only value supported here is *PKCS12V3 to indicate that only PKCS12 version 3 files will be used for importing and exporting entire certificate stores.

Export file password
INPUT; CHAR(*)

The password of the export file.

Length of export file password
INPUT; Binary(4)

The length of the password of the export file.

CCSID of export file password
INPUT; Binary(4)

This parameter is the CCSID of the export file password. If the value is 0, the default CCSID of the job will be used.

Error code
OUTPUT; 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
CPFB001 E	One or more input parameters is NULL or missing.
CPFB002 E	Certificate store does not exist.
CPFB003 E	Certificate store password is not valid.
CPFB004 E	User not authorized to certificate store.
CPFB005 E	Export file already exists.
CPFB006 E	An error occurred. The error code is &1.
CPFB007 E	User not authorized to directory or file.
CPFB008 E	The format name for the certificate store is not valid.
CPFB009 E	The format name for the export or import file is not valid.
CPFB00A E	Option &2 of the operating system is required to work with certificates.
CPF22F0 E	Unexpected errors occurred during processing.

API introduced: V5R3

Top | Security APIs | APIs by category

Find Certificate User (QSYFNDCU,QsyFindCertificateUser) API

Required Parameter Group for QSYFNDCU:

1	Certificate	Input	Char(*)
2	Type	Input	Binary(4)
3	Length of certificate	Input	Binary(4)
4	User profile	Output	Char(10)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyFindCertificateUser:

```
#include <qsydigid.h>

void QsyFindCertificateUser(
                  char    *Certificate,
```

```
int      Type,
int      Length_of_certificate,
char    *User_profile,
void    *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Find Certificate User (OPM, QSYFNDCU; ILE, QsyFindCertificateUser) API finds the user that is associated with a certificate.

Authorities and Locks

None.

Required Parameter Group

Certificate
INPUT; CHAR(*)

The certificate or certificate handle that is used to find the name of the user profile that has the certificate or certificate handle associated with it. This is not a text string.

Type INPUT; BINARY(4)

The type of the certificate.

The possible types are:

1 Entire X.509 public key certificate in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) encoding.
2 Certificate handle of the X.509 certificate.
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate. The type parameter indicates what this length refers to.

User profile
OUTPUT; CHAR(10)

The name of the user profile that is associated with the certificate. This field remains blank if the certificate is not found.

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
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF227D E	Certificate is not found.

Message ID	Error Message Text
CPF3BFF E	Required option &1 is not available.
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.
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.
CPF4AB9 E	User certificate function not successful.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Generate and Sign User Certificate Request (QYCUGSUC) API

Required Parameter Group:

1	User name	Input	Char(*)
2	Organization	Input	Char(*)
3	Organization unit	Input	Char(*)
4	City	Input	Char(*)
5	State	Input	Char(*)
6	Country or region	Input	Char(*)
7	Public key	Input	Char(*)
8	E-mail address	Input	Char(*)
9	File to store signed certificate	Input	Char(*)

Returned Value:

	Return code	Output	Binary(4)

Default Public Authority: *USE
Threadsafe: No

The Generate and Sign User Certificate Request (QYCUGSUC) API generates a user certificate request and then signs the certificate request using the local Certificate Authority (CA). The request to generate and sign the user certificate request must come from a Netscape, or compatible, browser session. The call to this program must be made using the DTW_DIRECTCALL language environment in Net.Data®.

Error information is returned as a return value from this program. The error code value can be captured using the RETURNS keyword on the function definition that uses DTW_DIRECTCALL.

Authorities and Locks

User Profile Authority
Caller of this API must have *ALLOBJ and *SECADM special authorities

API Public Authority
*USE

Required Parameter Group

User name
INPUT; CHAR(*)

The name of the user for which the certificate request was made. This is a required field.

Organization
INPUT; CHAR(*)

The organization information for the user. This is a required field.

Organization unit
INPUT; CHAR(*)

The organization unit information for the user. This may be a NULL string.

City INPUT; CHAR(*)

The city information for the user. This may be a NULL string.

State INPUT; CHAR(*)

The state information for the user. This is a required field.

Country or region
INPUT; CHAR(*)

The country or region information for the user. This is a required field.

Public key
INPUT; CHAR(*)

The public key for the certificate request. This value is generated using the ″keygen″ HTML directive. This is a required field.

E-mail address
Input; CHAR(*)

The e-mail address for the user. This may be a NULL string.

File to store signed certificate
Input; CHAR(*)

The absolute pathname for the file in which the signed certificate is stored. The file will be created if it does not exist. If the file already exists, the contents of the file will be replaced. This is a required field.

This parameter is assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Return Codes

Message ID	Error Message Text
0	Certificate was successfully signed.
-99	Unexpected error.
71	Unable to allocate storage.
93	The local Certificate Authority (CA) does not exist. Use Digital Certificate Manager (DCM) to create the local CA.
95	The password for the Local Certificate Authority (CA) certificate store is not stashed. Use DCM to change the password for the Local CA certificate store.
3843	The state value is too short. It must be at least 3 characters.
3845	The caller of this API does not have *ALLOBJ and *SECADM special authorities.
3857	The organization value is required.
3859	The country or region value is not valid. It must be 2 characters.
3956	The local CA does not allow creation of user certificates. You must change the policy data for the local CA using DCM.
4003	Certificate to be signed is not valid.

Example

The following is an example of a function call to this program using Net.Data.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 84.

```
%function(DTW_DIRECTCALL) signcert(IN  CHAR(10)   userName,
                                   IN  CHAR(64)   orgName,
                                   IN  CHAR(64)   orgUnitName,
                                   IN  CHAR(128)  city,
                                   IN  CHAR(128)  state,
                                   IN  CHAR(2)    countryRegion,
                                   IN  CHAR(1024) publicKey,
                                   IN  CHAR(128)  email,
                                   IN  CHAR(128)  storeFile) RETURNS(retVal) {
    %EXEC { /QSYS.LIB/QICSS.LIB/QYCUGSUC.PGM %}
%}
```

API introduced: V5R2

Top | Security APIs | APIs by category

Get Default Key Item (QYKMGDKI, QykmGetDefaultKeyItem)

Required Parameter Group:

1	Certificate store path and file name	Input	Char(*)
2	Length of certificate store path and file name	Input	Binary(4)
3	Format of certificate store path and file name	Input	Char(8)
4	Default certificate label	Output	Char(*)
5	Length of default certificate label provided	Input	Binary(4)
6	Length of default certificate label returned	Output	Binary(4)
7	Error code	I/O	Char(*)

Service Program Name: QYKMSYNC
Default Public Authority: *USE
Threadsafe: No

The Get Default Key Item API (OPM, QYKMGDKI; ILE, QykmGetDefaultKeyItem) allows a user to extract the label of the default certificate in a certificate store. If there is no assigned default certificate, no label is returned. **Note:** Option 34, Digital Certificate Manager, must be installed in order to use this API.

Authorities and Locks

Authority Required

To use this API, option 34 must be installed.

For the file object:

- *R authority to the certificate store.

For the directory:

- *X authority to the directory containing the certificate store.

Also, see the open()—Open File API for the authority needed to the certificate store.

Locks Object will be locked shared read.

Required Parameter Group

Certificate store path and file name
INPUT; Char(*)

The path name of the certificate store (kdb) you want to access. This path and file name may be absolute (i.e., entire path name) or relative to the current directory. If you are using format OBJN0100 (see below), this parameter is assumed to be represented in the coded character set identifier (CCSID) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Length of certificate store path and file name
INPUT; Binary(4)

The length of the certificate store path and file name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of certificate store path and file name
INPUT; CHAR(8)

The format of the certificate store path and file name parameter.

OBJN0100	The certificate store path and file name is a simple path name.
OBJN0200	The certificate path and file name is an LG-type path name.

Default certificate label
INPUT; CHAR(*)

The label of the default certificate in the certificate store. If there is no assigned default certificate, no label is returned. Also, if the length of the default certificate label provided (see next parameter) is not big enough to hold the label, the label is not returned.

Length of the default certificate label provided
INPUT; Binary(4)

The length provided for the label of the default certificate in the certificate store. This must be big enough to hold the certificate label. If not, the label is not returned, and the length of the default certificate label returned (see next parameter) will contain the minimum value that should be provided.

Length of the default certificate label returned
OUTPUT; Binary(4)

The actual length of the label of the default certificate in the certificate store. If this value is greater than 0 and no label is returned, then this value is the minimum value that should be provided on the length of the default certificate label provided (see previous parameter).

Error code
OUTPUT; 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
CPF180C E	Function &1 not allowed.
CPFB001 E	One or more input parameters is NULL or missing.
CPFB002 E	Certificate store does not exist.
CPFB004 E	User not authorized to certificate store.

Message ID	Error Message Text
CPFB006 E	An error occurred. The error code is &1.
CPFB007 E	User not authorized to directory or file.
CPFB008 E	The format name for the certificate store is not valid.
CPFB00A E	Option &2 of the operating system is required to work with certificates.
CPF22F0 E	Unexpected errors occurred during processing.

API introduced: V5R4

Import Certificate Store (QYKMIMPK, QykmImportKeyStore)

Required Parameter Group:

1	Certificate store path and file Name	Input	Char(*)
2	Length of certificate store path and file Name	Input	Binary(4)
3	Format of certificate store path and file Name	Input	Char(8)
4	Certificate store password	Input	Char(*)
5	Length of certificate store password	Input	Binary(4)
6	CCSID of certificate store password	Input	Binary(4)
7	Import path and file name	Input	Char(*)
8	Length of import path and file name	Input	Binary(4)
9	Format of import path and file name	Input	Char(8)
10	Version of import file	Input	Char(10)
11	Import file password	Input	Char(*)
12	Length of import file password	Input	Binary(4)
13	CCSID of import file password	Input	Binary(4)
14	Error code	I/O	Char(*)

Service Program Name: QYKMSYNC
Default Public Authority: *USE
Threadsafe: No

The Import Certificate Store API (OPM, QYKMIMPK; ILE, QykmImortKeyStore) allows a user to import an entire certificate store from a PKCS12 version 3 standard file. This allows for the import of private keys as well as record labels. Records with duplicate labels and/or public keys are not imported. **Note:** Option 34, Digital Certificate Manager, must be installed in order to use this API.

Authorities and Locks

Authority Required

To use this API, option 34 must be installed. You must also provide the password for the certificate store and know the password of the import file name.

For the file objects:

- *RW authority to the certificate store.
- *R authority to the import file.

For the directories:

- *WX authority to the directory containing the certificate store.
- *X authority to each directory in the paths to both the certificate store and import files.

Also, see the open()—Open File API for the authority needed to the certificate store and the import file.

Locks Object will be locked shared read.

Required Parameter Group

Certificate store path and file name
INPUT; Char(*)

The path name of the certificate store (kdb) to which you want to import. This path and file name may be absolute (i.e., entire path name) or relative to the current directory. If the file does not exist, it will be created. If you are using format OBJN0100 (see below), this parameter is assumed to be represented in the coded character set identifier (CCSID) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Length of certificate store path and file name
INPUT; Binary(4)

The length of the certificate store path and file name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of certificate store path and file name
INPUT; CHAR(8)

The format of the certificate store path and file name parameter.

OBJN0100 The certificate store path and file name is a simple path name.
OBJN0200 The certificate path and file name is an LG-type path name.

Certificate store password
INPUT; CHAR(*)

The password of the certificate store whose certificates will be imported from the given import file.

Length of certificate store password
INPUT; Binary(4)

The length of the password of the certificate store.

CCSID of certificate store password
INPUT; Binary(4)

This parameter is the CCSID of the certificate store password. If the value is 0, the default CCSID of the job will be used.

Import path and file name
INPUT; CHAR(*)

The path (including the name) of the import file from which all of the certificates are to be imported into the certificate store. This path and file name may be absolute (i.e., entire path name) or relative to the current directory. If you are using format OBJN0100 (see below), this parameter is assumed to be represented in the coded character set identifier (CCSID) currently in effect for the job. If the CCSID of the job is 65535, this parameter is assumed to be represented in the default CCSID of the job.

Length of import path and file name
INPUT; Binary(4)

The length of the import path and file name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of import path and file name
INPUT; CHAR(8)

The format of the import path and file name parameter.

OBJN0100	The import path and file name is a simple path name.
OBJN0200	The import path and file name is an LG-type path name.

Version of import file
INPUT; Char(10)

Currently, the only value supported here is *PKCS12V3 to indicate that only PKCS12 version 3 files will be used for importing and exporting entire certificate stores.

Import file password
INPUT; CHAR(*)

The password of the import file.

Length of import file password
INPUT; Binary(4)

The length of the password to the import file.

CCSID of import file password
INPUT; Binary(4)

This parameter is the CCSID of the import file password. If the value is 0, the default CCSID of the job will be used.

Error code
OUTPUT; 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
CPFB001 E	One or more input parameters is NULL or missing.
CPFB006 E	An error occurred. The error code is &1.
CPFB007 E	User not authorized to directory or file.
CPFB008 E	The format name for the certificate store is not valid.
CPFB009 E	The format name for the export or import file is not valid.
CPFB00A E	Option &2 of the operating system is required to work with certificates.
CPFB010 E	Import file does not exist.
CPFB011 E	Import file password is not valid.
CPFB012 E	Duplicate key exists.
CPF22F0 E	Unexpected errors occurred during processing.

API introduced: V5R3

Top | Security APIs | APIs by category

List User Certificates (QSYLSTUC, QsyListUserCertificates) API

Required Parameter Group for QSYLSTUC:

1	Qualified user space name	Input	Char(20)
2	User name	Input	Char(*)
3	Format name	Input	Char(8)
4	Selection control	Input	Char(*)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyListUserCertificates:

```
#include <qsydigid.h>

void QsyListUserCertificates
       (char           *Qualified_user_space_name,
        void           *User_name,
        char           *Format_name,
        char           *Selection_control,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The List User Certificates (OPM, QSYLSTUC; ILE, QsyListUserCertificates) API lists the certificates that are associated with the user profile. The generated list replaces any existing list in the user space.

A common scenario is that only one certificate is associated with an i5/OS® user profile at any given time, but more than one certificate may be associated with the same i5/OS user profile if each certificate is unique. The same certificate is not allowed to be associated with more than one i5/OS user profile.

Because certificates vary in length, the actual number of certificates that can be returned using the List User Certificates API will also vary. The total length of all of the certificates that have been added and the size of the user space determine the actual number that can be returned. In general, if more than a few hundred certificates are associated with an i5/OS user profile partial results may be returned when attempting to use the List User Certificates API to list the certificates. In addition to this maximum that varies due to certificate lengths, the List User Certificates API will not list more than 1000 certificates per user profile, no matter how small the certificates are for the user profile.

Selection control pairs that the caller may specify to do additional processing of the list may be useful for a user space that is smaller than the maximum size of a user space when the caller does not have authority to change the size of the user space. If more certificates are associated with an i5/OS user profile than can be returned by the List User Certificates API, the information status field in the generic header is set to indicate that the results are partial or incomplete.

Authorities and Locks

User Profile Authority

*USE

If *ALL is specified for the user profile name, the caller of this API must have *ALLOBJ special authority

If an EIM identifier is specified for the user profile name, the caller of this API must have *ALLOBJ special authority

User Space Authority
*CHANGE

User Space Library Authority
*EXECUTE

Required Parameter Group

Qualified user space name
INPUT; CHAR(20)

The name of the existing user space used to return the list of user certificates. The first 10 characters specify the user space name, and the second 10 characters specify the library.

You can use these special values for the library name:

**CURLIB*	The current library is used to locate the user space. If there is no current library, QGPL (general purpose library) is used.
**LIBL*	The library list is used to locate the user space.

User name
INPUT; CHAR(*)

The name of the user profile or the Enterprise Identity Mapping (EIM) identifier.

The following are valid selections:

**CURRENT*	The user profile that is currently running. The value must be 10 characters, blank padded.
**ALL*	All user profiles on this system. The value must be 10 characters, blank padded.
user profile	The name of the user profile. The value must be 10 characters, blank padded.
EIM identifier	To specify an EIM identifier for this parameter, the data must have the following format: *char(8)* The special value *EIMID. *binary(4)* The hex length of the EIM identifier. *char(*)* The EIM identifier.

Format name
INPUT; CHAR(8)

The content and format of the information that is returned for each certificate in the list data section of the qualified user space name.

The possible format names are:

"Certificate Format CERT0100 (ASN.1)" on page 24	Certificates in ASN.1 format
"Certificate Format CERT0200 (Plain Text)" on page 24	Certificates in plain text format

Selection control
INPUT; CHAR(*)

The structure that contains strings of interest and is used to limit which certificates are returned. For the format of this structure, see "Selection Control" on page 26.

Error code
I/O; CHAR(*)

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

Format

The certificate list generated in the user space consists of the following:

- A user area
- A generic header
- An input parameter section
- A list data section

In the generic header, the offset and length of the header section are set to zero because the header section is not used. The list data section has variable length entries, so the size of each entry is set to 0 in the generic header. For details about the user area and generic header, including which field indicates the number of entries returned or the offset to the first entry, see User spaces. For details about the formats in the list data section, see "Certificate Format CERT0100 (ASN.1)" on page 24 and "Certificate Format CERT0200 (Plain Text)" on page 24.

For details about the remaining items, see the following sections. For descriptions of each field in the list returned, see "Field Descriptions" on page 26.

Input Parameter Section

Offset		Type	Field
Dec	**Hex**		
0	0	BINARY(4)	Bytes returned in the returned records feedback information
0	0	CHAR(10)	User space name
10	A	CHAR(10)	User space library name
20	14	CHAR(10)	User name
30	1E	CHAR(8)	Format name
38	26	CHAR(2)	Reserved
40	28	BINARY(4)	Offset to selection control
44	2C	BINARY(4)	Offset to EIM identifier
48	30	BINARY(4)	Length of EIM identifier
The offset to this selection control is specified in a previous offset variable.		BINARY(4)	Length of selection control
		BINARY(4)	Number of selection pairs
		ARRAY(*) of BINARY(4)	Displacements to selection pairs
These fields repeat for each selection pair specified.		BINARY(4)	Length of selection pair
		CHAR(20)	Selection name
		ARRAY(*) of CHAR	Selection value
		CHAR(*)	EIM identifier

List Data Section

The list data section consists of certificates that are all set to one of the following formats as specified in the call to the API. The generic header has the number of list entries field.

Certificate Format CERT0100 (ASN.1)

The CERT0100 format consists of a certificate handle and the entire certificate encoded in ASN.1 DER (Abstract Syntax Notation 1 Distinguished Encoding Rules) format. The fields specified by the offsets and lengths in this format are not text fields.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Returned length of this certificate and format information
4	4	BINARY(4)	Available length of this certificate and format information
8	8	BINARY(4)	Offset to certificate handle
12	C	BINARY(4)	Length of certificate handle
16	10	BINARY(4)	Offset to ASN.1 format certificate
20	14	BINARY(4)	Length of ASN.1 format certificate
24	18	BINARY(4)	Offset to EIM identifier
28	1C	BINARY(4)	Length of EIM identifier
32	20	BINARY(4)	Offset to EIM local registry name
36	24	BINARY(4)	Length of EIM local registry name
40	28	BINARY(4)	Offset to user name
44	2C	BINARY(4)	Length of user name
		ARRAY(*) of CHAR	Fields specified by their offsets and lengths above

Certificate Format CERT0200 (Plain Text)

The CERT0200 format consists of a certificate handle and some of the sections of the certificate parsed into a more readable format. A field with a offset of 0 indicates that the field does not have a corresponding set of characters for the field value. A field length of 0 indicates that the field is empty, that it is not used in the certificate, or that it is not recognized. The fields specified by the offsets and lengths in this format are not all text fields.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Returned length of this certificate and format information
4	4	BINARY(4)	Available length of this certificate and format information
8	8	BINARY(4)	Offset to certificate handle
12	C	BINARY(4)	Length of certificate handle
16	10	BINARY(4)	Offset to version
20	14	BINARY(4)	Length of version
24	18	BINARY(4)	Offset to serial number
28	1C	BINARY(4)	Length of serial number
32	20	BINARY(4)	Offset to issuer's common name
36	24	BINARY(4)	Length of issuer's common name
40	28	BINARY(4)	Offset to issuer's country or region
44	2C	BINARY(4)	Length of issuer's country or region
48	30	BINARY(4)	Offset to issuer's state or province
52	34	BINARY(4)	Length of issuer's state or province

Offset		Type	Field
Dec	Hex		
56	38	BINARY(4)	Offset to issuer's locality
60	3C	BINARY(4)	Length of issuer's locality
64	40	BINARY(4)	Offset to issuer's organization
68	44	BINARY(4)	Length of issuer's organization
72	48	BINARY(4)	Offset to issuer's organizational unit
76	4C	BINARY(4)	Length of issuer's organizational unit
80	50	BINARY(4)	Offset to issuer's postal code
84	54	BINARY(4)	Length of issuer's postal code
88	58	BINARY(4)	Offset to validity period start
92	5C	BINARY(4)	Length of validity period start
96	60	BINARY(4)	Offset to validity period end
100	64	BINARY(4)	Length of validity period end
104	68	BINARY(4)	Offset to subject's common name
108	6C	BINARY(4)	Length of subject's common name
112	70	BINARY(4)	Offset to subject's country or region
116	74	BINARY(4)	Length of subject's country or region
120	78	BINARY(4)	Offset to subject's state or province
124	7C	BINARY(4)	Length of subject's state or province
128	80	BINARY(4)	Offset to subject's locality
132	84	BINARY(4)	Length of subject's locality
136	88	BINARY(4)	Offset to subject's organization
140	8C	BINARY(4)	Length of subject's organization
144	90	BINARY(4)	Offset to subject's organizational unit
148	94	BINARY(4)	Length of subject's organizational unit
152	98	BINARY(4)	Offset to subject's postal code
156	9C	BINARY(4)	Length of subject's postal code
160	A0	BINARY(4)	Offset to subject's public key algorithm
164	A4	BINARY(4)	Length of subject's public key algorithm
168	A8	BINARY(4)	Offset to issuer's unique ID (Version 2)
172	AC	BINARY(4)	Length of issuer's unique ID (Version 2)
176	B0	BINARY(4)	Offset to subject's unique ID (Version 2)
180	B4	BINARY(4)	Length of subject's unique ID (Version 2)
184	B8	BINARY(4)	Offset to issuer's e-mail address
188	BC	BINARY(4)	Length of issuer's e-mail address
192	C0	BINARY(4)	Offset to subject's e-mail address
196	C4	BINARY(4)	Length of subject's e-mail address
200	C8	BINARY(4)	Offset to EIM identifier
204	CC	BINARY(4)	Length of EIM identifier
208	D0	BINARY(4)	Offset to EIM local registry name
212	D4	BINARY(4)	Length of EIM local registry name

Offset		Type	Field
Dec	Hex		
216	D8	BINARY(4)	Offset to user name
220	DC	BINARY(4)	Length of user name
		ARRAY(*) of CHAR	Certificate information fields

Selection Control

The criteria is used to select or match certificates based on specified information.

This parameter is useful to reduce the total number of certificates that are returned in the list. The list of certificates is generated with only the specific selections that are of interest.

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

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of selection control
4	4	BINARY(4)	Number of selection pairs
8	8	ARRAY(*) of BINARY(4)	Displacements to selection pairs
These fields repeat for each selection pair specified		BINARY(4)	Length of selection pair
		CHAR(20)	Selection name
		ARRAY(*) of CHAR	Selection value

Field Descriptions

Available length of this certificate and format information. The available length of this certificate and format information. If this length is more than the returned length of this certificate and format information field, then not all of the fields were returned.

Certificate information fields. The actual data in the certificate. Specific fields can be accessed by using the offset to that specific field.

Displacements to selection pairs. An array of displacements to selection pairs from the beginning of the selection control.

EIM identifier. The EIM identifier that was specified on the call to the API.

Format name. The format of the returned output.

Length of ASN.1 format certificate. The length of the ASN.1 DER format certificate. This length refers to a field of hexadecimal bytes.

Length of certificate handle. The length of the certificate handle. This length refers to a field of hexadecimal bytes.

Length of EIM identifier. The length of the EIM identifier that was specified on the call to the API or to which the certificate is associated.

Length of EIM local registry name. The length of the EIM local registry name. This registry would be the target registry for the user name's association to the EIM identifier.

Length of issuer's common name. The length of the field that indicates the issuer's common name.

Length of issuer's country or region. The length of the field that indicates the issuer's country or region.

Length of issuer's e-mail address. The length of the field that indicates the issuer's e-mail address.

Length of issuer's locality. The length of the field that indicates the issuer's locality.

Length of issuer's organization. The length of the field that indicates the issuer's organization.

Length of issuer's organizational unit. The length of the field that indicates the issuer's organizational unit.

Length of issuer's postal code. The length of the field that indicates the issuer's postal code.

Length of issuer's state or province. The length of the field that indicates the issuer's state or province.

Length of issuer's unique ID (Version 2). The length of the field that indicates the issuer's unique ID (Version 2). This length refers to a field of hexadecimal bytes.

Length of selection control. The total number of bytes for the length itself, the bytes for the number of selection pairs, and the bytes for the array of displacements. It also includes the sum of the lengths of the selection pairs. The length of the selection control will vary due to the array of displacements and the selection pairs. A length of zero indicates that no selection control pairs are specified.

Length of selection pair. The length of the selection name and selection value fields and the bytes for the length itself. The length of the selection pair will vary due to the selection value. Valid values that are used are 24 bytes or larger. A value of 24 corresponds to a selection value that is empty and means that certificates should be returned when the corresponding value in the certificate is also empty or not recognized.

Length of serial number. The length of the field that indicates the serial number.

Length of subject's common name. The length of the field that indicates the subject's common name.

Length of subject's country or region. The length of the field that indicates the subject's country or region.

Length of subject's e-mail address. The length of the field that indicates the subject's e-mail address.

Length of subject's locality. The length of the field that indicates the subject's locality.

Length of subject's organization. The length of the field that indicates the subject's organization.

Length of subject's organizational unit. The length of the field that indicates the subject's organizational unit.

Length of subject's postal code. The length of the field that indicates the subject's postal code.

Length of subject's public key algorithm. The length of the field that indicates the subject's public key algorithm.

Length of subject's state or province. The length of the field that indicates the subject's state or province.

Length of subject's unique ID (Version 2). The length of the field that indicates the subject's unique ID (Version 2). This length refers to a field of hexadecimal bytes.

Length of user name. The length of the field that indicates the user name to which the certificate is associated.

Length of validity period start. The length of the field that indicates the beginning date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Length of validity period end. The length of the field that indicates the ending date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Length of version. The length of the field that indicates the version. This length refers to a field of hexadecimal bytes.

Number of selection pairs. The number of separate selection pairs in the generated list of certificates. All of the selection pairs must be satisfied for each certificate that is returned. If the number of selection pairs is 0, then all certificates are returned. The maximum allowed number of selection pairs is defined as QSY_MAX_SEL_NAMES.

Offset to ASN.1 format certificate. The offset to the ASN.1 DER format certificate. This offset refers to a field of hexadecimal bytes.

Offset to certificate handle. The offset to the certificate handle. This offset refers to a field of hexadecimal bytes.

Offset to EIM identifier. The offset to the EIM identifier that was specified on the call to the API or to which the certificate is associated.

Offset to EIM local registry name. The offset to the EIM local registry name.

Offset to issuer's common name. The offset to the field that indicates the issuer's common name.

Offset to issuer's country or region. The offset to the field that indicates the issuer's country or region.

Offset to issuer's e-mail address. The offset to the field that indicates the issuer's e-mail address.

Offset to issuer's locality. The offset to the field that indicates the issuer's locality.

Offset to issuer's organization. The offset to the field that indicates the issuer's organization.

Offset to issuer's organizational unit. The offset to the field that indicates the issuer's organizational unit.

Offset to issuer's postal code. The offset to the field that indicates the issuer's postal code.

Offset to issuer's state or province. The offset to the field that indicates the issuer's state or province.

Offset to issuer's unique ID (Version 2). The offset to the field that indicates the issuer's unique ID (Version 2). This offset refers to a field of hexadecimal bytes.

Offset to selection control. The offset to the selection control. The first field of the selection control is the length of selection control.

Offset to serial number. The offset to the field that indicates the serial number.

Offset to subject's common name. The offset to the field that indicates the subject's common name.

Offset to subject's country or region. The offset to the field that indicates the subject's country or region.

Offset to subject's e-mail address. The offset to the field that indicates the subject's e-mail address.

Offset to subject's locality. The offset to the field that indicates the subject's locality.

Offset to subject's organization. The offset to the field that indicates the subject's organization.

Offset to subject's organizational unit. The offset to the field that indicates the subject's organizational unit.

Offset to subject's postal code. The offset to the field that indicates the subject's postal code.

Offset to subject's public key algorithm. The offset to the field that indicates the subject's public key algorithm.

Offset to subject's state or province. The offset to the field that indicates the subject's state or province.

Offset to subject's unique ID (Version 2). The offset to the field that indicates the subject's unique ID (Version 2). This offset refers to a field of hexadecimal bytes.

Offset to user name. The offset to the user name to which the certificate is associated.

Offset to validity period start. The offset to the field that indicates the beginning date of the validity period.

Offset to validity period end. The offset to the field that indicates the ending date of the validity period.

Offset to version. The offset to the field that indicates the version. This offset refers to a field of hexadecimal bytes.

Reserved. An ignored field.

Returned length of this certificate and format information. The total length of this certificate and format information that was returned. This length is for one certificate.

Selection name. The selection that is used to limit which certificates from the validation list are returned. Selections indicate which fields of the certificate are to be examined for matching selection values. Selection names cannot be specified more than once. Selection names are defined with length QSY_SELCTRL_NAME_LEN.

Valid selection names are:

COMMONNAME	Client's common name
COUNTRY	Country or region in which the client resides
LOCALITY	Locality in which the client resides
STATEORPROVINCE	State or province in which the client resides
ORGANIZATION	Organization of the client
ORGANIZATIONALUNIT	Organizational unit of the client

PUBLICKEY	Public key of the certificate. This value is not text. It is the entire public key information as found in the certificate in ASN.1 DER format and it includes the tags and lengths. The actual public key found in the certificate is compared with the specified selection value that corresponds with this selection name. It is not returned in the list data section when the CERT0200 format name is specified.
EXPIRATIONDAYS	Certificates that are expired or will expire in the specified number of days. This value will be the number of days in character format (zoned decimal).
CERTIFICATEHANDLE	Handle for the certificate.

Selection value. The array of characters that is used for matching the corresponding field of the certificate. A match in the certificate indicates that the certificate is of interest. If the certificate does not contain matching characters in its corresponding field, the certificate will not be returned as part of the list. The length of the selection value can be determined by subtracting the fixed lengths of the selection name field and the length field from the length of selection pair. The comparison of the fields is done in the CCSID of the job and is case sensitive.

Example values:

John Smith
US
NY
XYZ Data
Security, Inc.
Secure Server
Certification
Authority

For example, to limit the certificates that are returned to only certificates that have US for the country or region, use the available definitions such as the 20-character name field defined by QSY_COUNTRY to indicate the following values in the selection control:

Length of
selection control:
38
Number of
selection pairs: 1
Displacement to
selection pair: 12

The corresponding selection pair for this example would use the following values:

Length of
selection pair: 26
Selection name:
COUNTRY
Selection value:
US

For another example, to indicate that all certificates that are found are to be returned, the selection control could indicate that there are no selection pairs to be used either by specifying that the length of the selection control is 0, and no selection pairs value will be checked, or by specifying that the number of selection pairs is 0 as follows:

Length of
selection control:
8
Number of
selection pairs: 0

User name. The name of the user profile that is specified on the call to the API. If this field contains *EIMID, then the Offset to EIM identifier and Length of EIM identifier fields can be used to determine the EIM identifier value that was specified on the call to the API.

User space library name. The library that contains the user space, as specified in the call to the API.

User space name. The name of the user space.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF2204 E	User profile &1 not found.
CPF2213 E	Not able to allocate user profile &1.
CPF2217 E	Not authorized to user profile &1.
CPF222E E	&1 special authority is required.
CPF2222 E	Storage limit is greater than specified for user profile &1.
CPF227B E	Certificate is not correct for the specified type.
CPF227E E	Selection control is not valid.
CPF3BFF E	Required option &1 is not available.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF4AB9 E	User certificate function not successful.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

List Validation List Certificates (QSYLSTVC,QsyListVldlCertificates) API

Required Parameter Group for QSYLSTVC:

1	Qualified user space name	Input	Char(20)
2	Validation list path name	Input	Char(*)
3	Length of path	Input	Binary(4)
4	Format name	Input	Char(8)
5	Selection control	Input	Char(*)

6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyListVldlCertificates:

```
#include <qsydigid.h>

void QsyListVldlCertificates
       (char           *Qualified_user_space_name,
        char           *Validation_list_path_name,
        int             Length_of_path,
        char           *Format_name,
        char           *Selection_control,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The List Validation List Certificates (OPM, QSYLSTVC; ILE, QsyListVldlCertificates) API lists the certificates in the validation list. The generated list replaces any existing list in the user space.

There may be many certificates in a validation list. Because a user space has a defined maximum length, there may be more certificates in a validation list than can be put into the user space. The List Validation List Certificates API allows the caller to specify additional selection processing so that only the certificates in the validation list which have fields matching the caller's selections are to be listed in the user space. The information status field in the generic header is set to indicate if the results are complete or not.

Authorities and Locks

Validation List Authority
*USE

Validation List Library Authority
*Execute

User Space Authority
*CHANGE

User Space Library Authority
*USE

Required Parameter Group

Qualified user space name
INPUT; CHAR(20)

The name of the existing user space used to return the list of validation list certificates. The first 10 characters specify the user space name, and the second 10 characters specify the library.

You can use these special values for the library name:

*CURLIB	The current library is used to locate the user space. If there is no current library, QGPL (general purpose library) is used.
*LIBL	The library list is used to locate the user space.

Validation list path name
INPUT; CHAR(*)

The fully qualified path name of the validation list.

Length of path
INPUT; BINARY(4)

The length of the validation list path name.

Format name
INPUT; CHAR(8)

The content and format of the information that is returned for each certificate in the list data section of the qualified user space name.

The possible formats are:

"Certificate Format CERT0100 (ASN.1)" on page 24	Certificates in Abstract Syntax Notation 1 (ASN.1) format
"Certificate Format CERT0200 (Plain Text)" on page 24	Certificates in plain text format

Selection control
INPUT; CHAR(*)

The structure that contains strings which are used to limit which certificates are returned. For the format of the structure, see "Selection Control" on page 26.

Error code
I/O; CHAR(*)

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

Usage Notes

In the list data section, the Offset to EIM identifier, Length of EIM identifier, Offset to EIM local registry name, Length of EIM local registry name, Offset to user name, and Length of user name fields will be 0.

Format

The certificate list generated in the user space consists of:

- A user area
- A generic header
- An input parameter section
- A list data section

In the generic header, the offset and length of the header section are set to zero because the header section is not used. The list data section has variable length entries, so the size of each entry is set to 0 in the generic header. For details about the user area and generic header, including which field indicates the number of entries returned or the offset to the first entry, see User spaces.

For details about the formats in the list data section, see "Certificate Format CERT0100 (ASN.1)" on page 24 and "Certificate Format CERT0200 (Plain Text)" on page 24.

For details about the remaining items, see the following sections. For descriptions of each field in the list returned, see "Field Descriptions" on page 34.

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	BINARY(4)	Offset to validation list path name
24	18	BINARY(4)	Length of validation list path name
28	1C	CHAR(8)	Format name
36	24	BINARY(4)	Offset to selection control
		CHAR(*)	Validation list path name
The offset to this selection control is specified in a previous offset variable.		BINARY(4)	Length of selection control
		BINARY(4)	Number of selection pairs
		ARRAY(*) of BINARY(4)	Displacements to selection pairs
These fields repeat for each selection pair specified.		BINARY(4)	Length of selection pair
		CHAR(20)	Selection name
		ARRAY(*) of CHAR	Selection value

Field Descriptions

Displacements to selection pairs. An array of displacements to selection pairs from the beginning of the selection control.

Format name. The format of the returned output.

Length of selection control. The total number of bytes for the length itself, for the number of selection pairs, and for the array of displacements. It also includes the sum of the lengths of the selection pairs. The length of the selection control will vary due to the array of displacements and the selection pairs. A length of zero is one of the ways to indicate that no selection control pairs are specified.

Length of selection pair. The total length of the selection name and selection value fields and the bytes for the length itself. The length of the selection pair will vary due to the selection value. Valid values that are used are 24 or larger. A value of 24 corresponds to a selection value that is empty and means that certificates should be returned when the corresponding value in the certificate is also empty or not recognized.

Length of validation list path name. The length of the path name of the validation list that is specified in the call to the API.

Number of selection pairs. The number of separate selection pairs in the generated list of certificates. All of the selection pairs must be satisfied for each certificate that is returned. If the number of selection pairs is 0, then all certificates are returned.

Offset to selection control. The offset to the selection control. The first field of the selection control is the length of selection control.

Offset to validation list path name. The offset to the full path name of the validation list that is specified in the call to the API.

Selection name. The selection that is used to limit which certificates from the validation list are returned. Selections made here indicate which field of the certificate is to be examined for a matching selection value. Selection names cannot be specified more than once. Selection names are defined with length QSY_SELCTRL_NAME_LEN.

Valid selection names are:

COMMONNAME	Client's common name
COUNTRY	Country or region in which the client resides
LOCALITY	Locality in which the client resides
STATEORPROVINCE	State or province in which the client resides
ORGANIZATION	Organization of the client
ORGANIZATIONALUNIT	Organizational unit of the client
PUBLICKEY	Public key of the certificate. This value is not text. It is the entire public key information as found in the certificate in ASN.1 DER format and it includes the tags and lengths. The actual public key found in the certificate is compared with the specified selection value that corresponds with this selection name. It is not returned in the list data section when the CERT0200 format name is specified.
EXPIRATIONDAYS	Certificates that are expired or will expire in the specified number of days. This value will be the number of days in character format (zoned decimal).
CERTIFICATEHANDLE	Handle for the certificate.

Selection value. The array of characters that is used for matching the corresponding field of the certificate. A match in the certificate indicates that the certificate is of interest. If the certificate does not contain matching characters in its corresponding field, the certificate will not be returned as part of the list. The length of the selection value can be determined by subtracting the fixed lengths of the selection name field and the length field from the length of selection pair. The comparison of the fields is done in the CCSID of the job and is case sensitive.

User space library name specified. The library that contains the user space, as specified in the call to the API.

User space name specified. The name of the user space.

Validation list path name. The path name of the validation list.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPFA09C E	Not authorized to object.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227B E	Certificate is not correct for the specified type.
CPF227E E	Selection control is not valid.
CPF3BFF E	Required option &1 is not available.
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.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.

Message ID	Error Message Text
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Open List of User Certificates (QSYOLUC) 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	User name	Input	Char(10)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

The Open List of User Certificates (QSYOLUC) API provides a list of user certificates associated with a user.

Authorities and Locks

User Profile Authority
*USE

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. For more information about the format of the data returned in the receiver variable, see "Certificate Format CERT0100 (ASN.1)" on page 24 and "Certificate Format CERT0200 (Plain Text)" on page 24.

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. For formats that contain variable length data, the receiver variable length must be large enough to hold the fixed portion of the record.

List information
OUTPUT; CHAR(80)

The variable that is used to return status information about the list of user certificates that were opened. See "Format of List Information" on page 37 for a description of this parameter.

Number of records to return
INPUT; BINARY(4)

The number of records containing a user certificate to return. The value -1 indicates that all the records containing user certificates should be returned.

Format name
INPUT; CHAR(8)

The name of the format that is used to list user certificates.

The possible format names are:

"Certificate Format CERT0100 (ASN.1)" on page 24	Certificates in ASN.1 format in ASCII encoding.
"Certificate Format CERT0200 (Plain Text)" on page 24	Certificates in plain text format. Character fields are encoded in the job CCSID.

User name
INPUT; CHAR(10)

The name of the user profile.

The following is a valid selection:

**CURRENT*	The user profile that is currently running.

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 List Information

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Total records
4	4	BINARY(4)	Records returned
8	8	CHAR(4)	Request handle
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
44	2C	CHAR(36)	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 The date, in YYMMDD (year, month, and day) format.
8-13 The time of day, in HHMMSS (hours, minutes, and seconds) format.

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

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

Possible values follow:

I Incomplete information. An interruption causes 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 are read and returned.

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

List status indicator. The status of building the list.

The possible value follows:

2 The list has been completely built.

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

- The number of records that will fit into the receiver variable.
- The number of records in the list.
- The number of records that are 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. An ignored field.

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

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
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.
CPF227E E	Selection control is not valid.
CPF3BFF E	Required option &1 is not available.

Message ID	Error Message Text
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF4AB9 E	User certificate function not successful.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R3

Top | Security APIs | APIs by category

Parse Certificate (QSYPARSC, QsyParseCertificate) API

Required Parameter Group for QSYPARSC:

1	Certificate	Input	Char(*)
2	Type	Input	Binary(4)
3	Length of certificate	Input	Binary(4)
4	Format	Input	Char(8)
5	Receiver variable	Output	Char(*)
6	Length of receiver variable	Input	Binary(4)
7	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyParseCertificate:

```
#include <qsydigid.h>

void QsyParseCertificate(
                  char    *Certificate,
                  int      Type,
                  int      Length_of_certificate,
                  char    *Format_name,
                  char    *Receiver_variable,
                  int      Length_of_receiver_variable,
     void    *Error_code
     );
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Parse Certificate (OPM, QSYPARSC; ILE, QsyParseCertificate) API parses a certificate and returns the results to the caller.

Authorities and Locks

None.

Required Parameter Group

Certificate
INPUT; CHAR(*)

The entire certificate encoded in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) format. This is not a text string.

Type INPUT; BINARY(4)

The type or format of the certificate.

The possible types are:

1 Entire X.509 public key certificate in ASN.1 DER encoding.
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate.

Format
INPUT; CHAR(8)

The format of the parsed certificate.

The possible types are:

"Certificate Format CERT0200 (Plain Text)" on page 24	All text fields available.
"Certificate Format CERT0210" on page 41	All text fields available. None of the fields are translated from the ASCII format that they had in the certificate into the job CCSID.

Receiver variable
OUTPUT; CHAR(*)

The storage that is provided by the user to hold the certificate text. For more information, see "Format of Receiver Variable."

Length of receiver variable
INPUT; BINARY(4)

The length of the storage that is provided by the user.

Error code
I/O; CHAR(*)

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

Usage Notes

If format CERT0200 is specified, the Offset to EIM identifier, Length of EIM identifier, Offset to EIM local registry name, Length of EIM local registry name, Offset to user name, and Length of user name fields will be 0.

Format of Receiver Variable

For details about the format that is returned in the receiver variable, for Format CERT0200 see "Certificate Format CERT0200 (Plain Text)" on page 24.

The following tables describe the order and format of the data returned in the receiver variable for Format CERT0210. For detailed descriptions of the fields in the tables, see "Field Descriptions" on page 42.

Note: A distinguished name (DN) consists of the following fields in the order presented:

- Common name
- Organizational unit
- Organization
- Locality
- State
- Postal code
- Country or region

Certificate Format CERT0210

The CERT0210 format consists of a certificate handle and some of the sections of the certificate parsed into a more readable format. If the length of a field is 0 or the offset to a field is 0, then the field does not contain any information. Either the field is empty, it is not used in the certificate, or it is not recognized. The fields specified by the offsets and lengths in this format are either text or hexadecimal bytes as indicated in the field descriptions.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Returned length of this certificate and format information
4	4	BINARY(4)	Available length of this certificate and format information
8	8	BINARY(4)	Offset to certificate handle
12	C	BINARY(4)	Length of certificate handle
16	10	BINARY(4)	Offset to version
20	14	BINARY(4)	Length of version
24	18	BINARY(4)	Offset to serial number
28	1C	BINARY(4)	Length of serial number
32	20	BINARY(4)	Offset to issuer's common name
36	24	BINARY(4)	Length of issuer's common name
40	28	BINARY(4)	Offset to issuer's country or region
44	2C	BINARY(4)	Length of issuer's country or region
48	30	BINARY(4)	Offset to issuer's state or province
52	34	BINARY(4)	Length of issuer's state or province
56	38	BINARY(4)	Offset to issuer's locality
60	3C	BINARY(4)	Length of issuer's locality
64	40	BINARY(4)	Offset to issuer's organization
68	44	BINARY(4)	Length of issuer's organization
72	48	BINARY(4)	Offset to issuer's organizational unit
76	4C	BINARY(4)	Length of issuer's organizational unit
80	50	BINARY(4)	Offset to issuer's postal code
84	54	BINARY(4)	Length of issuer's postal code
88	58	BINARY(4)	Offset to validity period start

Offset		Type	Field
Dec	Hex		
92	5C	BINARY(4)	Length of validity period start
96	60	BINARY(4)	Offset to validity period end
100	64	BINARY(4)	Length of validity period end
104	68	BINARY(4)	Offset to subject's common name
108	6C	BINARY(4)	Length of subject's common name
112	70	BINARY(4)	Offset to subject's country or region
116	74	BINARY(4)	Length of subject's country or region
120	78	BINARY(4)	Offset to subject's state or province
124	7C	BINARY(4)	Length of subject's state or province
128	80	BINARY(4)	Offset to subject's locality
132	84	BINARY(4)	Length of subject's locality
136	88	BINARY(4)	Offset to subject's organization
140	8C	BINARY(4)	Length of subject's organization
144	90	BINARY(4)	Offset to subject's organizational unit
148	94	BINARY(4)	Length of subject's organizational unit
152	98	BINARY(4)	Offset to subject's postal code
156	9C	BINARY(4)	Length of subject's postal code
160	A0	BINARY(4)	Offset to subject's public key algorithm
164	A4	BINARY(4)	Length of subject's public key algorithm
168	A8	BINARY(4)	Offset to issuer's unique ID (Version 2)
172	AC	BINARY(4)	Length of issuer's unique ID (Version 2)
176	B0	BINARY(4)	Offset to subject's unique ID (Version 2)
180	B4	BINARY(4)	Length of subject's unique ID (Version 2)
184	B8	BINARY(4)	Offset to issuer's email address
188	BC	BINARY(4)	Length of issuer's email address
192	C0	BINARY(4)	Offset to subject's email address
196	C4	BINARY(4)	Length of subject's email address
216	D8	BINARY(4)	Offset to issuer's distinguished name (DN) in DER representation
220	DC	BINARY(4)	Length of issuer's distinguished name (DN) in DER representation
224	E0	BINARY(4)	Offset to subject's distinguished name (DN) in DER representation
228	E4	BINARY(4)	Length of subject's distinguished name (DN) in DER representation
232	E8	BINARY(4)	Offset to certificate public key in DER representation
236	EC	BINARY(4)	Length of certificate public key in DER representation
		ARRAY(*) of CHAR	Certificate information

Field Descriptions

Available length of this certificate and format information. The available length of this certificate and format information. If this length is more than the returned length of this certificate and format information field, then not all of the fields were returned.

Fields specified by their offsets and lengths above. The fields that were specified by their offsets and lengths prior to this field.

Certificate information. The actual data in the certificate. Specific fields can be accessed by using the offset to the specific field.

Format name. The format of the returned output.

Length of ASN.1 format certificate. The length of the ASN.1 DER format certificate. This length refers to a field of hexadecimal bytes.

Length of certificate handle. The length of the certificate handle. This length refers to a field of hexadecimal bytes.

Length of certificate public key in DER representation. The length of the certificate public key. This length refers to a field of hexadecimal bytes.

Length of issuer's common name. The length of the field that indicates the issuer's common name.

Length of issuer's country or region. The length of the field that indicates the issuer's country or region.

Length of issuer's distinguished name (DN) in DER representation. The length of the field that indicates the issuer's DN in DER representation.

Length of issuer's email address. The length of the field that indicates the issuer's email address.

Length of issuer's locality. The length of the field that indicates the issuer's locality.

Length of issuer's organization. The length of the field that indicates the issuer's organization.

Length of issuer's organizational unit. The length of the field that indicates the issuer's organizational unit.

Length of issuer's postal code. The length of the field that indicates the issuer's postal code.

Length of issuer's state or province. The length of the field that indicates the issuer's state or province.

Length of issuer's unique ID (Version 2). The length of the field that indicates the issuer's unique ID (Version 2). This length refers to a field of hexadecimal bytes.

Length of serial number. The length of the field that indicates the serial number.

Length of subject's common name. The length of the field that indicates the subject's common name.

Length of subject's country or region. The length of the field that indicates the subject's country or region.

Length of subject's distinguished name (DN) in DER representation. The length of the field that indicates the subject's DN in DER representation.

Length of subject's email address. The length of the field that indicates the subject's email address.

Length of subject's locality. The length of the field that indicates the subject's locality.

Length of subject's organization. The length of the field that indicates the subject's organization.

Length of subject's organizational unit. The length of the field that indicates the subject's organizational unit.

Length of subject's postal code. The length of the field that indicates the subject's postal code.

Length of subject's public key algorithm. The length of the field that indicates the subject's public key algorithm.

Length of subject's state or province. The length of the field that indicates the subject's state or province.

Length of subject's unique ID (Version 2). The length of the field that indicates the subject's unique ID (Version 2). This length refers to a field of hexadecimal bytes.

Length of validity period start. The length of the field that indicates the beginning date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Length of validity period end. The length of the field that indicates the ending date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Length of version. The length of the field that indicates the version. This length refers to a field of hexadecimal bytes.

Offset to ASN.1 format certificate. The offset to the ASN.1 DER format certificate. This offset refers to a field of hexadecimal bytes.

Offset to certificate handle. The offset to the certificate handle. This offset refers to a field of hexadecimal bytes.

Offset to certificate public key in DER representation. The offset to the certificate public key. This offset refers to a field of hexadecimal bytes.

Offset to issuer's common name. The offset to the field that indicates the issuer's common name.

Offset to issuer's country or region. The offset to the field that indicates the issuer's country or region.

Offset to issuer's distinguished name (DN) in DER representation. The offset to the field that indicates the issuer's DN in DER representation.

Offset to issuer's email address. The offset to the field that indicates the issuer's email address.

Offset to issuer's locality. The offset to the field that indicates the issuer's locality.

Offset to issuer's organization. The offset to the field that indicates the issuer's organization.

Offset to issuer's organizational unit. The offset to the field that indicates the issuer's organizational unit.

Offset to issuer's postal code. The offset to the field that indicates the issuer's postal code.

Offset to issuer's state or province. The offset to the field that indicates the issuer's state or province.

Offset to issuer's unique ID (Version 2). The offset to the field that indicates the issuer's unique ID (Version 2). This offset refers to a field of hexadecimal bytes.

Offset to serial number. The offset to the field that indicates the serial number.

Offset to subject's common name. The offset to the field that indicates the subject's common name.

Offset to subject's country or region. The offset to the field that indicates the subject's country or region.

Offset to subject's distinguished name (DN) in DER representation. The offset to the field that indicates the subject's DN in DER representation.

Offset to subject's email address. The offset to the field that indicates the subject's email address.

Offset to subject's locality. The offset to the field that indicates the subject's locality.

Offset to subject's organization. The offset to the field that indicates the subject's organization.

Offset to subject's organizational unit. The offset to the field that indicates the subject's organizational unit.

Offset to subject's postal code. The offset to the field that indicates the subject's postal code.

Offset to subject's public key algorithm. The offset to the field that indicates the subject's public key algorithm.

Offset to subject's state or province. The offset to the field that indicates the subject's state or province.

Offset to subject's unique ID (Version 2). The offset to the field that indicates the subject's unique ID (Version 2). This offset refers to a field of hexadecimal bytes.

Offset to validity period start. The offset to the field that indicates the beginning date of the validity period.

Offset to validity period end. The offset to the field that indicates the ending date of the validity period.

Offset to version. The offset to the field that indicates the version. This offset refers to a field of hexadecimal bytes.

Reserved. An ignored field.

Returned length of this certificate and format information. The total length of this certificate and format information that was returned. This length is for one certificate and can be used to access the next certificate in the list.

User name. The name of the user profile that is specified in the call to the API.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF3BFF E	Required option &1 is not available.
CPF3CF1 E	Error code parameter not valid.

Message ID	Error Message Text
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF3C1D E	Length specified in parameter &1 not valid.
CPF3C1E E	Required parameter &1 omitted.
CPF3C21 E	Format name &1 is not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was 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

Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API

Required Parameter Group for QSYRGAP:

1	Application ID	Input	Char(*)
2	Length of application ID	Input	Binary(4)
3	Application controls	Input	Char(*)
4	Error Code	I/O	Char(*)

Default Public Authority: *EXCLUDE
Threadsafe: Yes

Syntax for QsyRegisterAppForCertUse:

```
#include <qsyrgap1.h>

void QsyRegisterAppForCertUse
       (char               *Application_ID,
        int                *Length_of_application_ID,
        Qsy_App_Controls_T  *Application_controls,
        void               *Error_code);
```

Service Program: QSYRGAP1
Default Public Authority: *EXCLUDE
Threadsafe: Yes

The Register Application For Certificate Use (OPM, QSYRGAP; ILE, QsyRegisterAppForCertUse) API registers an application with the registration facility. The application controls provide additional information needed to define the application.

You can update an application entry by reregistering the application (using the replace control key) with new values for the application control keys.

The application type control key is set the first time the application is registered and cannot be changed.

When an application is registered, the registration information is stored using the Registration Facility.

Authorities and Locks

API Public Authority
*EXCLUDE

Registration Lock
*EXCL

Required Parameter Group

Application ID
INPUT; CHAR(*)

The application ID to register. IBM-supplied i5/OS® applications are named QIBM_*ccc_name*, where *ccc* is the component identifier. User-supplied application IDs should not preface their application ID with QIBM. User-supplied application IDs should start with the company name to eliminate most problems that involve unique names. Application IDs should use an underscore (_) to separate parts of the name (for example, QIBM_I5OS_HOSTSERVER). Also, IDs for related applications should start with the same name (for example, QIBM_DIRSRV_SERVER and QIBM_DIRSRV_REPLICATION).

The first character of the application ID must be one of the following:

A-Z	Uppercase A-Z

The remaining characters in the application ID must be made up of the following characters:

A-Z	Uppercase A-Z
0-9	Digits 0-9
.	Period
_	Underscore

Length of application ID
INPUT; BINARY(4)

The length of the specified application ID. The length must be a value from 1 to 100. If the application type is 4 (object signing application), then the length must be a value from 1 to 30.

Application controls
INPUT; CHAR(*)

The application control fields for defining the application. Any field not specified will be given the default value. The information must be in the following format:

Number of variable length records	BINARY(4) The total number of all of the variable length records.
Variable length records	The fields of the application controls to set. Refer to "Format for Variable Length Record" for more information.

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 Variable Length Record

The following table shows the layout of the variable length record. For a detailed description of each field, see "Field Descriptions" on page 48.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Length of variable length record
4	4	BINARY(4)	Application control key
8	8	BINARY(4)	Length of data
12	C	CHAR(*)	Data

If the length of the data is longer than the key field's data length, the data is truncated at the right. No message is issued.

If the length of the data is shorter than the key field's data length and the key contains binary data, an error message is issued. If the key does not contain binary data, the field is padded with blanks.

It is not an error to specify a key more than once. If duplicate keys are specified, the last specified value for that key is used.

Each variable length record must be 4-byte aligned. If not, unpredictable results may occur.

Refer to "Application Control Keys" for more information about the valid values for these fields.

Field Descriptions

Application control key. The application control to be set. Refer to the "Key" column in the "Application Control Keys" table for the list of valid control keys.

Data. The value to which a specific application control is to be set.

Length of data. The length of the application control value.

Length of variable length record. The length of the record, including this field.

Application Control Keys

The following table shows the valid application control keys for the application control key field of the variable length record. For a detailed description of each field, see "Field Descriptions."

Key	Type	Field
1	CHAR(20)	Qualified exit program name
2	CHAR(50)	Application description
3	CHAR(27)	Qualified message file name and message identifier for application description
4	CHAR(1)	Limit CA certificates trusted
5	CHAR(1)	Replace
6	CHAR(1)	Threadsafe
7	CHAR(1)	Multithreaded job action
8	CHAR(1)	Application type
9	CHAR(10)	Application user profile
10	CHAR(1)	Client authentication supported
11	CHAR(1)	Client authentication required
12	CHAR(1)	Perform certificate revocation processing

Field Descriptions

Application description. The text for the application description. When this key is specified, the qualified message file name and message identifier for application description key must not be specified. The default value is blanks.

Application type. The type of application. This control is set when the application is registered and cannot be changed. The default value is 1. Valid values for this key are:

1 Server application. A server application provides a service for another process on the system, host, or network.

2 Client application. A client application requests a service from another process on the system, host, or network.

4 Object signing application. This application is used when signing objects. The application ID for this application can be specified on the Sign Object (QYDOSGNO) API. When an object signing application is registered, a corresponding function is registered with the same ID (see Register Function (QSYRGFN, QsyRegisterFunction) API). A user must have access to the corresponding function to sign objects using this application ID. By default, only users with *ALLOBJ special authority will have access to the corresponding function.

Application user profile. The user profile associated with the application. This is the user profile under which the application runs. If a user profile name is specified, then the specified user profile is given access to the QIBM_QSY_SYSTEM_CERT_STORE function (see Register Function (QSYRGFN, QsyRegisterFunction) API). This function gives the specified user profile access to the *SYSTEM certificate store without having to be authorized to the actual object, but only when using the certificate associated with the application to establish a secure session. The default value is *NONE. The following special value may be specified:

**NONE* No user profile will be associated with the application. This value must be specified if the application type is 4 (object signing application).

Client authentication required. Whether client authentication is required. The default value is 0.

0 No client authentication is done. This value must be specified if the client authentication supported value is 0.

1 Client authentication is required. The client is authenticated as part of the SSL handshake protocol processing. During the SSL handshake processing, the server requests a certificate from the client. The certificate must be valid and must be signed by a Certificate Authority (CA) that the server recognizes and trusts. If the client does not have a valid certificate, then the server ends the SSL handshake and does not establish an SSL session between the client and server.

Client authentication supported. Whether the application supports client authentication. The default value is 0.

0 The application does not support client authentication. If this value is specified, the client authentication required value must be 0. This value must be specified if the application type is 2 (client application) or 4 (object signing application).

1 The application supports client authentication.

Limit CA certificates trusted. Whether the application trusts all of the CA certificates that are trusted in the *SYSTEM certificate store or a subset of the CA certificates. A client application uses the list of trusted CA certificates to validate the peer certificate that is sent to the application. A server application that supports client authentication uses the list of trusted CA certificates to validate the certificate that is sent from the client. The default value is 1.

0 The application trusts all the CA certificates that are trusted in the *SYSTEM certificate store. This value must be specified if the application type is 4 (object signing application). This value is recommended for server applications that do not support client authentication.

1 The application trusts a subset of the list of CA certificates that are trusted in the *SYSTEM certificate store. If this value is specified, the system administrator must specify which of the CA certificates that are trusted in the *SYSTEM certificate store also are trusted by the application. Otherwise, the application will not trust any of the CA certificates. Using Digital Certificate Manager (DCM), the system administrator can add and remove CA certificates from the list of trusted CA certificates for the application. The application must be a client application or a server application that supports client authentication to be able to use DCM to manage the list of CA certificates that the application trusts.

Multithreaded job action. The action to take in a multithreaded job. This key has no direct relationship with the threadsafe key; however, the value for the threadsafe key can be used to determine the multithreaded job action. The default value is 0. Valid values for this key are:

0 Use the QMLTTHDACN system value to determine the action to take.
1 Run the exit program in a multithreaded job.
2 Run the exit program in a multithreaded job and send informational message CPI3C80.
3 Do not run the exit program in a multithreaded job and send informational message CPI3C80.

If you do use the threadsafe value to determine the value for the multithreaded job action, consider the following recommendations:

1. If the threadsafe value is 0, the multithreaded job action should be set to 3.
2. If the threadsafe value is 1, the multithreaded job action should be set to 0.
3. If the threadsafe value is 2, the multithreaded job action should be set to 1.

Perform certificate revocation processing. Whether certificate revocation processing is performed when the certificate associated with the application is validated. The default value is 0.

0 Certificate revocation processing is not performed when the certificate associated with the application is validated. If the certificate has been revoked, it will still be considered valid.
1 Certificate revocation processing is performed when the certificate associated with the application is validated. If the certificate has been revoked, it will not be valid.

Qualified exit program name. The exit program name and library for the application. The first 10 characters contain the exit program name; the next 10 characters contain the library name in which the exit program resides. The exit program does not need to exist at registration time. A specific library name must be specified. The special values *LIBL and *CURLIB are not supported. The default value is program QSY_NOPGM in library QSY_NOLIB.

This exit program is called when a certificate is assigned to the application, an assigned certificate is changed, or an assigned certificate is removed. It is called when a Certificate Authority (CA) certificate is added to or removed from the list of trusted CA certificates for the application. It also is called when an attempt is made to deregister the application. The exit program can determine whether or not the application can be deregistered. This exit program also is called when the information for a registered application is updated. See “Digital Certificate Management Exit Programs” on page 76 for detailed information about the information that is passed to the exit program for each of the possible calls to the program.If the exit program is the default value, then it will not be called.

Qualified message file name and message identifier for application description. A message file and message identifier that contains the application description. When this key is specified, the application description key must not be specified. The message file and message identifier do not have to exist at the time of registration. The default value is blanks. See “Qualified Message File Format” on page 51 for the format of this field.

Replace. Whether to replace an existing registered application. The default value is 0.

0 Do not replace an existing registered application. If this value is specified and the application is already registered, the request will fail.

1 Replace an existing registered application. If this value is specified and the application is not already registered, the application will be registered. If the application is already registered, only the application control keys that are specified on this call are replaced. Any other application control keys that were previously specified will keep their values.

2 Replace an existing registered application, but do not replace application control keys that are controlled by a system administrator. If this value is specified and the application is not already registered, the application will be registered. If the application is already registered, only the application control keys that are specified on this call are replaced. Any other application control keys that were previously specified will keep their values. Application control keys that are controlled by a system administrator are not replaced, even if they are specified on this call. These application control keys include:

- Client authentication required
- Limit CA certificates trusted
- Perform certificate revocation processing

This value should be used by install exit programs to ensure that values set by the system adminstrator are not replaced by the install exit program.

Threadsafe. Whether the exit program entry is threadsafe. This key has no direct relationship with the multithreaded job action key. It is intended for documentation purposes only. The default value is 1. Valid values for this key are:

0 The exit program entry is not threadsafe.

1 The threadsafe status of the exit program entry is not known.

2 The exit program entry is threadsafe.

Qualified Message File Format

The following table shows the layout of the qualified message file name and message identifier for the application description field. For a detailed description of each field, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(10)	Message file name
10	A	CHAR(10)	Message file library name
20	14	CHAR(7)	Message identifier

Field Descriptions

Message file library name. The library name in which the message file resides. The special value *CURLIB is not supported. The possible values are:

**LIBL* Search the library list for the message file. This value uses the first message file in the library list that contains the message identifier.

library name The name of the message library in which the message file resides.

Message file name. The name of the message file that contains the application description.

Message identifier. The message identifier for the application description.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF2225 E	Not able to allocate internal system object.
CPF222E E	&1 special authority is required.
CPF220E E	Application &1 not registered.
CPF220F E	Application &1 already registered.
CPF229E E	Application ID &1 not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C4D E	Length &1 for key &2 not valid.
CPF3C81 E	Value for key &1 not valid.
CPF3C82 E	Key &1 not valid for API &2.
CPF3C83 E	Key &1 not allowed with value specified for key &2.
CPF3C84 E	Key &1 required with value specified for key &2.
CPF3C88 E	Number of variable length records &1 is not valid.
CPF3C90 E	Literal value cannot be changed.
CPF3CD9 E	Requested function cannot be performed at this time.
CPF3CDA E	Registration facility repository not available for use.
CPF3CF1 E	Error code parameter not valid.
CPF3CF2 E	Error(s) occurred during running of &1 API.
CPF8100 E	All CPF81xx messages could be returned. xx is from 01 to FF.
CPF9810 E	Library &1 not found.
CPF9811 E	Program &1 in library &2 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R4

Top | Security APIs | APIs by category

Remove User Certificate (QSYRMVUC, QsyRemoveUserCertificate) API

Required Parameter Group for QSYRMVUC:

1	User profile	Input	Char(*)
2	Certificate	Input	Char(*)
3	Type	Input	Binary(4)
4	Length of certificate	Input	Binary(4)
5	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyRemoveUserCertificate:

```
#include <qsydigid.h>

void QsyRemoveUserCertificate(
                  void    *User_profile,
                  char    *Certificate,
                  int      Type,
                  int      Length_of_certificate,
                  void    *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Remove User Certificate (OPM, QSYRMVUC; ILE, QsyRemoveUserCertificate) API removes a certificate from an i5/OS® user profile.

Authorities and Locks

User Profile Authority

If the user profile specified is not the current user for the job, the caller of this API must have *SECADM special authority and *USE and *OBJMGT authorities to the specified user profile.

If an EIM identifier is specified for the user profile name, the caller of this API must have *SECADM and *ALLOBJ special authority

Required Parameter Group

User profile

INPUT; CHAR(*)

The name of the user profile or the Enterprise Identity Mapping (EIM) identifier that holds the certificate. The following are valid selections:

**CURRENT* The user profile that is currently running. The value must be 10 characters, blank padded.

user profile The name of the user profile. The value must be 10 characters, blank padded.

EIM identifier To specify an EIM identifier for this parameter, the data must have the following format:

char(8) The special value *EIMID.

binary(4) The hex length of the EIM identifier.

char(*) The EIM identifier.

Certificate

INPUT; CHAR(*)

The certificate or handle of the certificate that identifies the entire certificate that is to be removed. This is not a text string.

Type INPUT; BINARY(4)

The type that identifies the contents in the certificate field.

The possible types are:

1 Entire X.509 public key certificate in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) encoding.

2 Certificate handle for X.509 certificate.

3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate

INPUT; BINARY(4)

The length of the certificate or handle of the certificate that was specified.

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
CPFA0AA E	Error occurred while attempting to obtain space.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF2204 E	User profile &1 not found.
CPF2213 E	Not able to allocate user profile &1.
CPF2217 E	Not authorized to user profile &1.
CPF222E E	&1 special authority is required.
CPF2222 E	Storage limit is greater than specified for user profile &1.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF227D E	Certificate is not found.
CPF3BFF E	Required option &1 is not available.
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.
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.
CPF4AB9 E	User certificate function not successful.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Remove Validation List Certificate (QSYRMVVC, QsyRemoveVldlCertificate) API

Required Parameter Group for QSYRMVVC:

1	Validation list path name	Input	Char(*)
2	Length of path	Input	Binary(4)
3	Certificate	Input	Char(*)
4	Type	Input	Binary(4)
5	Length of certificate	Input	Binary(4)
6	Error code	I/O	Char(*)

Default Public Authority: *USE
Threadsafe: Yes

Syntax for QsyRemoveVldlCertificate:

```
#include <qsydigid.h>

void QsyRemoveVldlCertificate(
                  char    *Validation_list_path_name,
                  int      Length_of_path,
                  char    *Certificate,
                  int      Type,
                  int      Length_of_certificate,
                  void    *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Remove Validation List Certificate (OPM, QSYRMVVC; ILE, QsyRemoveVldlCertificate) API removes a certificate from a validation list.

Authorities and Locks

Validation List Authority
*USE and *DLT

Validation List Library Authority
*EXECUTE

Required Parameter Group

Validation list path name
INPUT; CHAR(*)

The fully qualified path name of the validation list.

Length of path
INPUT; BINARY(4)

The length of the validation list path.

Certificate
INPUT; CHAR(*)

The certificate or handle of the certificate that identifies the entire certificate that is to be removed. This is not a text string.

Type INPUT; BINARY(4)

The type that identifies the contents in the certificate field.

The possible types are:

1 Entire X.509 public key certificate in Abstract Syntax Notation 1 Distinguished Encoding Rules (ASN.1 DER) encoding.
2 Certificate handle for X.509 certificate.
3 Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding. Note that the characters of the Base 64 encoding are the ASCII representation and not the EBCDIC representation.

Length of certificate
INPUT; BINARY(4)

The length of the certificate or certificate handle that was provided.

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
CPFA0AA E	Error occurred while attempting to obtain space.
CPFA09C E	Not authorized to object.
CPF1F41 E	Severe error occurred while addressing parameter list.
CPF227A E	Certificate type is not valid.
CPF227B E	Certificate is not correct for the specified type.
CPF227D E	Certificate is not found.
CPF3BFF E	Required option &1 is not available.

Message ID	Error Message Text
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.
CPF3C1E E	Required parameter &1 omitted.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9810 E	Library &1 not found.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V4R2

Top | Security APIs | APIs by category

Retrieve Certificate Information (QYCURTVCI, QycuRetrieveCertificateInfo) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format of certificate information	Input	Char(8)
4	Certificate store name	Input	Char(*)
5	Length of certificate store name	Input	Binary(4)
6	Format of certificate store name	Input	Char(8)
7	Certificate store password	Input	Char(*)
8	Length of certificate store password	Input	Binary(4)
9	CCSID of certificate store password	Input	Binary(4)
10	Selection control	Input	Char(*)
11	Error code	I/O	Char(*)

Program: QICSS/QYCURTVCI

Default Public Authority: *USE

Threadsafe: No

Syntax for QycuRetrieveCertificateInfo:

```
#include <qycucerti.h>

void QycuRetrieveCertificateInfo
       (void           *Receiver_variable,
  int            *Length_receiver_variable,
        char           *Format_certificate_info,
        char           *Certificate_store_name,
         int           *Length_certificate_store_name,
        char           *Format_certificate_store_name,
 char           *Certificate_store_password,
  int           *Length_certificate_store_password,
  int           *CCSID_certificate_store_password,
        char           *Selection_control,
        void           *Error_code);
```

Service Program: QICSS/QYCUCERTI
Default Public Authority: *USE
Threadsafe: No

The Retrieve Certificate Information (OPM, QYCURTVCI; ILE, QycuRetrieveCertificateInfo) API retrieves information from server or CA certificates. For example, you can retrieve information about certificates that are expiring within a given date range.

Authorities and Locks

Authority Required

The caller of this API must provide the password for the certificate store. In addition, the caller must have *ALLOBJ and *SECADM special authorities.

Locks Object will be locked shared read.

Required Parameter Group

Note: Do not use quotation marks in the input parameters.

Receiver variable
OUTPUT; CHAR(*)

The variable that is to receive the certificate information.

Length of receiver variable
INPUT; BINARY(4)

The length of the receiver variable. If the length specified is larger than the actual size of the receiver variable, the results will not be predictable. The minimum length is 8 bytes.

Format of certificate information
INPUT; CHAR(8)

The content and format of the information that is returned for each certificate is specified here.

The possible format names are:

"**RTCI0100 Format**" *on page 59* Certificate labels
"**RTCI0200 Format**" *on page 59* Certificate labels and expiration information
"**RTCI0300 Format**" *on page 59* All certificate information

Certificate store name
INPUT; CHAR(*)

The certificate store from which you want to retrieve the list of certificates. The following values can be used for the certificate store name:

**SYSTEM*	The *SYSTEM certificate store.
**OBJECTSIGNING*	The *OBJECTSIGNING certificate store.
**SIGNATUREVERIFICATION*	The *SIGNATUREVERIFICATION certificate store.
Directory path and file name	The fully qualified Integrated File System (IFS) directory path and file name of the certificate store. The directory path must start with a leading forward slash (/), for example, */mydirectory/mystore.kdb.* If you are using format OBJN0100, the path and file name are assumed to be represented in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, the path and file name are assumed to be represented in the default CCSID of the job.

Length of certificate store name
INPUT; Binary(4)

The length of the certificate store name. If the format specified is OBJN0200 (see below), this field must include the QLG path name structure length in addition to the length of the path name itself. If the format specified is OBJN0100 (see below), only the length of the path name itself is included.

Format of certificate store name
INPUT; CHAR(8)

The format of the certificate store path and file name parameter.

OBJN0100	The certificate store path and file name is a simple path name. If you are specifying *SYSTEM, *OBJECTSIGNING, or *SIGNATUREVERIFICATION for the certificate store name, use this format.
OBJN0200	The certificate path and file name is an LG-type path name.

Certificate store password
INPUT; CHAR(*)

The password for the certificate store.

Length of certificate store password
INPUT; Binary(4)

The length of the password of the certificate store.

CCSID of certificate store password
INPUT; Binary(4)

This parameter is the CCSID of the certificate store password. If the value is 0, the default CCSID of the job will be used.

Selection control
INPUT; CHAR(*)

The control information used to limit which certificates are returned. For the format of this structure, see "Selection Control" on page 65.

Error code
OUTPUT; CHAR(*)

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

Receiver Formats

The following tables describe the order and format of the data returned in a receiver variable. For detailed descriptions of each field, see "Receiver Field Descriptions" on page 61.

RTCI0100 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 certificate entry
12	C	BINARY(4)	Number of certificate entries returned
16	10	CHAR(*)	Reserved
Certificate entry information. These fields are repeated for each certificate entry returned.			
		BINARY(4)	Displacement to next certificate entry
		BINARY(4)	Displacement to certificate label
		BINARY(4)	Length of certificate label
		CHAR(*)	Certificate label

RTCI0200 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 certificate entry
12	C	BINARY(4)	Number of certificate entries returned
16	10	CHAR(*)	Reserved
Certificate entry information. These fields are repeated for each certificate entry returned.			
		BINARY(4)	Displacement to next certificate entry
		CHAR(14)	Validity period end
		CHAR(2)	Reserved
		BINARY(4)	Displacement to certificate label
		BINARY(4)	Length of certificate label
		BINARY(4)	Displacement to subject's common name
		BINARY(4)	Length of subject's common name
		ARRAY(*) of CHAR	Certificate information fields

RTCI0300 Format

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned

Offset		Type	Field
Dec	Hex		
4	4	BINARY(4)	Bytes available
8	8	BINARY(4)	Offset to first certificate entry
12	C	BINARY(4)	Number of certificate entries returned
16	10	CHAR(*)	Reserved
Certificate entry information. These fields are repeated for each certificate entry returned.			
		BINARY(4)	Displacement to next certificate entry
		CHAR(1)	Trusted status
		CHAR(1)	Private key indicator
		CHAR(1)	Key storage location
		CHAR(14)	Validity period start
		CHAR(14)	Validity period end
		CHAR(16)	Key usage extensions
		CHAR(11)	Reserved
		BINARY(4)	Key size
		BINARY(4)	Displacement to certificate label
		BINARY(4)	Length of certificate label
		BINARY(4)	Displacement to serial number
		BINARY(4)	Length of serial number
		BINARY(4)	Displacement to subject's common name
		BINARY(4)	Length of subject's common name
		BINARY(4)	Displacement to subject's country or region
		BINARY(4)	Length of subject's country or region
		BINARY(4)	Displacement to subject's state or province
		BINARY(4)	Length of subject's state or province
		BINARY(4)	Displacement to subject's locality
		BINARY(4)	Length of subject's locality
		BINARY(4)	Displacement to subject's organization
		BINARY(4)	Length of subject's organization
		BINARY(4)	Displacement to subject's organizational unit
		BINARY(4)	Length of subject's organizational unit
		BINARY(4)	Displacement to subject's postal code
		BINARY(4)	Length of subject's postal code
		BINARY(4)	Displacement to issuer's common name
		BINARY(4)	Length of issuer's common name
		BINARY(4)	Displacement to issuer's country or region
		BINARY(4)	Length of issuer's country or region
		BINARY(4)	Displacement to issuer's state or province
		BINARY(4)	Length of issuer's state or province
		BINARY(4)	Displacement to issuer's locality
		BINARY(4)	Length of issuer's locality

Offset		Type	Field
Dec	Hex		
		BINARY(4)	Displacement to issuer's organization
		BINARY(4)	Length of issuer's organization
		BINARY(4)	Displacement to issuer's organizational unit
		BINARY(4)	Length of issuer's organizational unit
		BINARY(4)	Displacement to issuer's postal code
		BINARY(4)	Length of issuer's postal code
		BINARY(4)	Displacement to CRL location
		BINARY(4)	Length of CRL location
		BINARY(4)	Displacement to LDAP server name
		BINARY(4)	Length of LDAP server name
		BINARY(4)	Displacement to private key label
		BINARY(4)	Length of private key label
		BINARY(4)	Displacement to IP address
		BINARY(4)	Length of IP address
		BINARY(4)	Displacement to domain name
		BINARY(4)	Length of domain name
		BINARY(4)	Displacement to email address
		BINARY(4)	Length of email address
		BINARY(4)	Displacement to first cryptographic device
		BINARY(4)	Number of cryptographic devices
		BINARY(4)	Number of cryptographic devices returned
		ARRAY(*) of CHAR	Certificate information fields
Cryptographic device information. These fields are repeated for each cryptographic device returned.			
		BINARY(4)	Displacement to next cryptographic device
		BINARY(4)	Displacement to cryptographic device name
		BINARY(4)	Length of cryptographic device name
		ARRAY(*) of CHAR	Cryptographic device information fields (names)

Receiver Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Certificate label. The label for the certificate. The label is returned in the CCSID (coded character set identifier) currently in effect for the job. If the CCSID of the job is 65535, the label is returned in the default CCSID of the job. The certificate label is a null terminated string.

Displacement to certificate label. The displacement from the beginning of the entry to the field that indicates the certificate label.

Displacement to CRL location. The displacement from the beginning of the entry to the field that indicates the CRL location.

Displacement to cryptographic device name. The displacement from the beginning of the entry to the field that indicates the cryptographic device name.

Displacement to domain name. The displacement from the beginning of the entry to the field that indicates the domain name.

Displacement to email address. The displacement from the beginning of the entry to the field that indicates the email address.

Displacement to first cryptographic device. The displacement from the beginning of the entry to the field that indicates the first cryptographic device.

Displacement to IP address. The displacement from the beginning of the entry to the field that indicates the IP address.

Displacement to issuer's common name. The displacement from the beginning of the entry to the field that indicates the issuer's common name.

Displacement to issuer's country or region. The displacement from the beginning of the entry to the field that indicates the issuer's country or region.

Displacement to issuer's locality. The displacement from the beginning of the entry to the field that indicates the issuer's locality.

Displacement to issuer's organization. The displacement from the beginning of the entry to the field that indicates the issuer's organization.

Displacement to issuer's organizational unit. The displacement from the beginning of the entry to the field that indicates the issuer's organizational unit.

Displacement to issuer's postal code. The displacement from the beginning of the entry to the field that indicates the issuer's postal code.

Displacement to issuer's state or province. The displacement from the beginning of the entry to the field that indicates the issuer's state or province.

Displacement to LDAP server name. The displacement from the beginning of the entry to the field that indicates the LDAP server name.

Displacement to next certificate entry. The displacement from the beginning of this entry to the next entry.

Displacement to next cryptographic device. The displacement from the beginning of the current cryptographic device entry to the next entry.

Displacement to private key label. The displacement from the beginning of the entry to the field that indicates the private key label.

Displacement to serial number. The displacement from the beginning of the entry to the field that indicates the serial number.

Displacement to subject's common name. The displacement from the beginning of the entry to the field that indicates the subject's common name.

Displacement to subject's country or region. The displacement from the beginning of the entry to the field that indicates the subject's country or region.

Displacement to subject's locality. The displacement from the beginning of the entry to the field that indicates the subject's locality.

Displacement to subject's organization. The displacement from the beginning of the entry to the field that indicates the subject's organization.

Displacement to subject's organizational unit. The displacement from the beginning of the entry to the field that indicates the subject's organizational unit.

Displacement to subject's postal code. The displacement from the beginning of the entry to the field that indicates the subject's postal code.

Displacement to subject's state or province. The displacement from the beginning of the entry to the field that indicates the subject's state or province.

Key size. The size of the key in bytes.

Key storage location A single character that indicates where the key is stored.

Possible values:

0 The key is stored is software
1 The key is stored in hardware
2 The key is stored in hardware encryption

Key usage extensions The key usage extension values for the certificate. If the certificate has the key usage extension, the field is 1. If not, the field is 0.

This field contains the following fields:

DigitalSignature CHAR(1)

Whether the certificate has the digital signature extension.

NonRepudiation CHAR(1)

Whether the certificate has the nonrepudiation extension.

KeyEncipherment CHAR(1)

Whether the certificate has the key encipherment extension.

DataEncipherment CHAR(1)

Whether the certificate has the data encipherment extension.

KeyAgreement CHAR(1)

Whether the certificate has the key agreement extension.

KeyCertSign CHAR(1)

Whether the certificate has the key certificate signature extension.

CRLSign CHAR(1)

Whether the certificate has the CRL signature extension.

EncipherOnly CHAR(1)

Whether the certificate has the encipher only extension.

DecipherOnly CHAR(1)

Whether the certificate has the decipher only extension.

Reserved CHAR(7)

An ignored field.

Length of certificate label. The length of the field that contains the certificate label.

Length of CRL location. The length of the field that indicates the CRL location.

Length of cryptographic device name. The length of the field that indicates the cryptographic device name.

Length of domain name. The length of the field that indicates the domain name.

Length of email address. The length of the field that indicates the email address.

Length of IP address. The length of the field that indicates the IP address.

Length of issuer's common name. The length of the field that indicates the issuer's common name.

Length of issuer's country or region. The length of the field that indicates the issuer's country or region.

Length of issuer's locality. The length of the field that indicates the issuer's locality.

Length of issuer's organization. The length of the field that indicates the issuer's organization.

Length of issuer's organizational unit. The length of the field that indicates the issuer's organizational unit.

Length of issuer's postal code. The length of the field that indicates the issuer's postal code.

Length of issuer's state or province. The length of the field that indicates the issuer's state or province.

Length of LDAP server name. The length of the field that indicates the LDAP server name.

Length of private key label. The length of the field that indicates the private key label. Will be 0 if the key storage location is 0.

Length of serial number. The length of the field that indicates the serial number.

Length of subject's common name. The length of the field that indicates the subject's common name.

Length of subject's country or region. The length of the field that indicates the subject's country or region.

Length of subject's locality. The length of the field that indicates the subject's locality.

Length of subject's organization. The length of the field that indicates the subject's organization.

Length of subject's organizational unit. The length of the field that indicates the subject's organizational unit.

Length of subject's postal code. The length of the field that indicates the subject's postal code.

Length of subject's state or province. The length of the field that indicates the subject's state or province.

Number of certificate entries returned. The number of certificate entries returned. If the receiver variable is not large enough to hold all of the information, this number contains only the number of certificate entries actually returned.

Number of cryptographic devices. The number of cryptographic devices returned.

Offset to first certificate entry. The offset to the first certificate entry returned. The offset is from the beginning of the structure. If no entries are returned, the offset is set to zero.

Private key indicator One character indicator that indicates if the certificate has a private key.

Possible values:

0 The certificate does not have a private key.
1 The certificate does have a private key.

Trusted status One character indicator that indicates if the certificate is trusted.

Possible values:

0 The certificate is not trusted.
1 The certificate is trusted.

Reserved. An ignored field.

Validity period start. The field that indicates the beginning date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Validity period end. The field that indicates the ending date of the validity period. The first 8 characters consist of 4 characters for the year, 2 characters for the month, and 2 characters for the day. The last 6 characters consist of 2 characters for the hours, 2 characters for the minutes, and 2 characters for the seconds.

Selection Control

The criteria is used to select or match certificates based on specified information.

This parameter is useful to reduce the total number of certificates that are returned in the list. The list of certificates is generated with only the specific selections that are of interest.

The following shows the format of the selection control parameter. For detailed descriptions of the fields in the table, see "Selection Control Field Descriptions" on page 66.

Offset		Type	Field
Dec	Hex		
0	0	BINARY(4)	Length of selection control
4	4	BINARY(4)	Number of selection pairs
8	8	ARRAY(*) of BINARY(4)	Offsets to selection pairs

Offset		Type	Field
Dec	Hex		
These fields repeat for each selection pair specified		BINARY(4)	Length of selection pair
		CHAR(20)	Selection name
		ARRAY(*) of CHAR	Selection value

Selection Control Field Descriptions

Length of selection control. The total number of bytes for the length itself, the bytes for the number of selection pairs, and the bytes for the array of displacements. It also includes the sum of the lengths of the selection pairs. The length of the selection control will vary due to the array of displacements and the selection pairs. A length of zero indicates that no selection control pairs are specified.

Number of selection pairs. The number of separate selection pairs in the generated list of certificates. All of the selection pairs must be satisfied for each certificate that is returned. If the number of selection pairs is 0, then all certificates are returned. The maximum allowed number of selection pairs is defined as QYCU_MAX_SEL_NAMES.

Length of selection pair. The length of the selection name and selection value fields and the bytes for the length itself. The length of the selection pair will vary due to the selection value. Valid values that are used are 24 bytes or larger.

Offsets to selection pairs. An array of offsets to selection pairs from the beginning of the selection control.

Selection name. The selection that is used to limit which certificates are returned. Selections indicate which fields of the certificate are to be examined for matching selection values. Selection names cannot be specified more than once.

Valid selection names are:

EXPIRATIONDAYS CHAR(4) Certificates that are expired or will expire in the specified number of days. This value will be the number of days in character format (zoned decimal). The valid range is from 1 to 365 days.

CERTIFICATETYPE CHAR(1) This may be server or CA.

Possible values:

0 Server certificate

1 CA certificate

CERTIFICATELABEL CHAR (*) Certificate whose label match the label specified. When choosing this selection criteria, the other selection criteria are not allowed.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attemption to obtain space.
CPFA0C1 E	CCSID &1 not valid.
CPFA049 E	Certificate store does not exist.
CPFA09C E	User not authorized to certificate store.
CPFB001 E	One or more input parameters is NULL or missing.
CPFB003 E	Certificate store password is not valid.
CPFB006 E	An error occurred. The error code is &1.

Message ID	Error Message Text
CPF222E E	&1 special authority is required.
CPF227E E	Selection control is 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.
CPF3C36 E	Number of parameters, &1, entered for this API was not valid.
CPF3C90 E	Literal value cannot be changed.

« API introduced: V6R1

QsyRetrieveDigitalIDConfig()—Retrieve Digital ID Configuration Information

Syntax for QsyRetrieveDigitalIDConfig:

```
#include <qsydigid.h>

void QsyRetrieveDigitalIDConfig
       (void           *Receiver_variable,
        int             Length_of_receiver_variable,
        char           *Format_name,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Retrieve Digital ID Configuration Information (QsyRetrieveDigitalIDConfig) API will retrieve digital ID configuration information. This is the information that defines the Lightweight Directory Access Protocol (LDAP) server for where to store digital certificates, and connection information for the server.

Authorities and Locks

QSYDIGID Validation List Object
*USE

QUSRSYS Library
*EXECUTE

Note: For the bind password to be returned, the user must have *USE, *ADD, and *UPD authorities to the validation 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 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. The minimum length is 8 bytes.

Format name
INPUT; CHAR(8)

The format of the configuration information to be returned.

The following format name may be used:

"RDCI0100 Format" Digital ID configuration information.

Error code
I/O; CHAR(*)

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

RDCI0100 Format

The following table describes the information that is returned in the receiver variable for the RDCI0100 format. For detailed descriptions of the fields, see "Field Descriptions" on page 69.

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	BINARY(4)	Bytes returned
4	4	BINARY(4)	Bytes available
8	8	CHAR(1)	Enabled indicator
9	9	CHAR(1)	Use SSL indicator
10	10	CHAR(2)	Reserved
12	C	BINARY(4)	Port number
16	10	BINARY(4)	Offset to server name
20	14	BINARY(4)	Length of server name
24	18	BINARY(4)	Offset to directory DN
28	1C	BINARY(4)	Length of directory DN
32	20	BINARY(4)	Offset to bind DN
36	24	BINARY(4)	Length of bind DN
40	28	BINARY(4)	Offset to bind password
44	2C	BINARY(4)	Length of bind password
		CHAR(*)	Server name
		CHAR(*)	Directory DN
		CHAR(*)	Bind DN
		CHAR(*)	Bind password

Field Descriptions

Bytes available. The number of bytes of data available to be returned. All available data is returned if enough space is provided.

Bytes returned. The number of bytes of data returned.

Bind DN. The Distinguished Name (DN) of the entry used when binding to the LDAP server.

Bind password. The password to use in association with the bind DN.

Directory DN. The DN for where in the LDAP server the user certificates are to be stored.

Enabled indicator. Specifies whether or not the configuration information is enabled for use.

0 The configuration information is not enabled. Digital certificates for users will be stored locally.
1 The configuration information is enabled. If Enterprise Identity Mapping (EIM) is configured and operational, then digital certificates for users will be stored in LDAP and the mapping from the certificate to a user profile will be stored in EIM.

Length of bind DN. The length of the field that contains the bind distinguished name (DN).

Length of bind password. The length of the field that contains the bind password.

Length of directory DN. The length of the field that contains the directory distinguished name (DN).

Length of server name. The length of the field that contains the server name.

Port number. The port number to use when connecting to the LDAP server.

Offset to bind DN. The offset to the field that contains the bind distinguished name (DN).

Offset to bind password. The offset to the field that contains the bind password.

Offset to directory DN. The offset to the field that contains the directory distinguished name (DN).

Offset to server name. The offset to the field that contains the server name.

Reserved. Reserved data.

Server name. The domain name of the LDAP server on which to store user certificates.

Use SSL indicator. Specifies whether or not Secure Sockets Layer (SSL) is used for secure access when connecting to the LDAP server.

0 A secure connection using SSL is not used when connecting to the LDAP server.
1 A secure connection using SSL is used when connecting to the LDAP server. Digital Certificate Manager (DCM) must be used to assign a certificate to the IBM® Directory Server client (QIBM_GLD_DIRSRV_CLIENT) application.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF22F0 E	Unexpected errors occurred during processing.

Message ID	Error Message Text
CPF3BFF E	Required option &1 is not available.
CPF3CF1 E	Error code parameter not valid.
CPF3C1D E	Length specified in parameter &1 not valid.
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.
CPF3C90 E	Literal value cannot be changed.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Top | Security APIs | APIs by category

QsySetDigitalIDConfig()—Set Digital ID Configuration Information

Syntax for QsySetDigitalIDConfig:

```
#include <qsydigid.h>

void QsySetDigitalIDConfig
       (char           *Configuration_information,
        int             Length_of_configuration_information,
        char           *Format_name,
        void           *Error_code);
```

Service Program: QSYDIGID
Default Public Authority: *USE
Threadsafe: Yes

The Set Digital ID Configuration Information (QsySetDigitalIDConfig) API will set digital ID configuration information. This is the information that defines the Lightweight Directory Access Protocol (LDAP) server for where to store digital certificates, and connection information for the server.

Authorities and Locks

QSYDIGID Validation List Object
*CHANGE

QUSRSYS Library
*EXECUTE

Required Parameter Group

Configuration information
INPUT; CHAR(*)

The configuration information that is being set. See “SDCI0100 Format” on page 71 for the definition of the fields for this parameter.

Length of configuration information
INPUT; BINARY(4)

The length of the configuration information. This area must be as large as the format specified.

Format name

INPUT; CHAR(8)

The format of the configuration information.

The following format name may be used:

"SDCI0100 Format" Digital ID configuration information.

Error code

I/O; CHAR(*)

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

SDCI0100 Format

The following table shows the information that must be specified in the configuration information parameter when format SDCI0100 is specified. For a detailed description of each field, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(1)	Enabled indicator
1	1	CHAR(1)	Use SSL indicator
2	2	CHAR(2)	Reserved
4	4	BINARY(4)	Port number
8	8	BINARY(4)	Offset to server name
12	C	BINARY(4)	Length of server name
16	10	BINARY(4)	Offset to directory DN
20	14	BINARY(4)	Length of directory DN
24	18	BINARY(4)	Offset to bind DN
28	1C	BINARY(4)	Length of bind DN
32	20	BINARY(4)	Offset to bind password
36	24	BINARY(4)	Length of bind password
		CHAR(*)	Server name
		CHAR(*)	Directory DN
		CHAR(*)	Bind DN
		CHAR(*)	Bind password

Field Descriptions

Bind DN. The Distinguished Name (DN) used when binding to the LDAP server. The DN that is specified must have sufficient LDAP authorization to create objects and modify objects in the specified directory DN.

Bind password. The password to use in association with the bind DN.

Directory DN. The DN for where in the LDAP server the user certificates are to be stored.

Enabled indicator. Specifies whether or not the configuration information is enabled for use.

blank The value does not change.
0 The configuration information is not enabled. Digital certificates for users will be stored locally.
1 The configuration information is enabled. If Enterprise Identity Mapping (EIM) is configured and operational, then digital certificates for users will be stored in LDAP and the mapping from the certificate to a user profile will be stored in EIM.

Length of bind DN. The length of the field that contains the bind distinguished name (DN).

-1 The current value is not changed.
0 The current value is removed.
1 - 1000 The current value is changed to the specified value.

Length of bind password. The length of the field that contains the bind password.

-1 The current value is not changed.
0 The current value is removed.
1 - 600 The current value is changed to the specified value.

Length of directory DN. The length of the field that contains the directory distinguished name (DN).

-1 The current value is not changed.
0 The current value is removed.
1 - 1000 The current value is changed to the specified value.

Length of server name. The length of the field that contains the server name.

-1 The current value is not changed.
0 The current value is removed.
1 - 1000 The current value is changed to the specified value.

Port number. The port number to use when connecting to the LDAP server. The suggested port number for non-secure access is 389. The suggested port number for secure access is 636. A value of -1 indicates that the port number does not change.

Offset to bind DN. The offset to the field that contains the bind distinguished name (DN). If the Length of bind DN is -1 or 0, then this value must be 0.

Offset to bind password. The offset to the field that contains the bind password. If the Length of bind password is -1 or 0, then this value must be 0.

Offset to directory DN. The offset to the field that contains the directory distinguished name (DN). If the Length of directory DN is -1 or 0, then this value must be 0.

Offset to server name. The offset to the field that contains the server name. If the Length of server name is -1 or 0, then this value must be 0.

Reserved. Reserved data. This value must be hexadecimal zero.

Server name. The domain name of the LDAP server on which to store user certificates.

Use SSL indicator. Specifies whether or not Secure Sockets Layer (SSL) is used for secure access when connecting to the LDAP server.

blank	The value does not change.
0	A secure connection using SSL is not used when connecting to the LDAP server.
1	A secure connection using SSL is used when connecting to the LDAP server. Digital Certificate Manager (DCM) must be used to assign a certificate to the IBM® Directory Server client (QIBM_GLD_DIRSRV_CLIENT) application.

Error Messages

Message ID	Error Message Text
CPFA0AA E	Error occurred while attempting to obtain space.
CPF22F0 E	Unexpected errors occurred during processing.
CPF226D E	Not all information stored.
CPF3BFF E	Required option &1 is not available.
CPF3CF1 E	Error code parameter not valid.
CPF3C21 E	Format name &1 is not valid.
CPF3C3B E	Value for parameter &2 for API &1 not valid.
CPF3C3C E	Value for parameter &1 not valid.
CPF3C39 E	Value for reserved field not valid.
CPF3C90 E	Literal value cannot be changed.
CPF9801 E	Object &2 in library &3 not found.
CPF9802 E	Not authorized to object &2 in &3.
CPF9803 E	Cannot allocate object &2 in library &3.
CPF9804 E	Object &2 in library &3 damaged.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API introduced: V5R3

Top | Security APIs | APIs by category

Sign User Certificate Request (QYCUSUC) API

Required Parameter Group:

1	Signed certificate	Output	Char(*)
2	Signed certificate length	Input	Binary(4)
3	Certificate request	Input	Char(*)
4	E-mail address	Input	Char(*)

Returned Value:

	Return code	Output	Binary(4)

Default Public Authority: *USE
Threadsafe: No

The Sign User Certificate Request (QYCUSUC) API signs a user certificate request using the local Certificate Authority (CA). The request to sign the user certificate request must come from an Internet Explorer, or compatible, browser session. The call to this program must be made using the DTW_DIRECTCALL language environment in Net.Data®.

Error information is returned as a return value from this program. The error code value can be captured using the RETURNS keyword on the function definition that uses DTW_DIRECTCALL.

Authorities and Locks

User Profile Authority
Caller of this API must have *ALLOBJ and *SECADM special authorities

API Public Authority
*USE

Required Parameter Group

Signed certificate
OUTPUT; CHAR(*)

The storage for returning the signed certificate. The signed certificate will be a NULL terminated string. This storage is allocated by Net.Data and is referenced using the environment variable that was specified on the call.

Signed certificate length
INPUT; BINARY(4)

The length of the storage provided by the signed certificate parameter.

Certificate request
INPUT; CHAR(*)

The certificate request data to sign. This should be the data that is returned from the Enroll.CreatePKCS10() call in Net.Data.

E-mail address
Input; CHAR(*)

The e-mail address for the user. This may be a NULL string.

Return Codes

Message ID	Error Message Text
0	Certificate was successfully signed.
-99	Unexpected error.
71	Unable to allocate storage. The certificate request data may not be valid.
93	The local Certificate Authority (CA) does not exist. Use Digital Certificate Manager (DCM) to create the local CA.
95	The password for the Local Certificate Authority (CA) certificate store is not stashed. Use DCM to change the password for the Local CA certificate store.
321	Signed certificate length is not large enough to return the signed certificate.
3845	The caller of this API does not have *ALLOBJ and *SECADM special authorities.
3956	The local CA does not allow creation of user certificates. You must change the policy data for the local CA using DCM.
4003	Certificate request to be signed is not valid.

Example

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 84.

The following is an example of a function call to this program using Net.Data from an Internet Explorer browser session. Note that the size specified for the second parameter must be the same as the number of characters allocated for the first parameter.

```
%function(DTW_DIRECTCALL) signcert(OUT CHAR(5000) signedCert,
                                   IN  INT        signedCertLen,
                                   IN  CHAR(4000) certData,
                                   IN  CHAR(128)  email) RETURNS(retVal) {
    %EXEC { /QSYS.LIB/QICSS.LIB/QYCUSUC.PGM %}
%}
```

The following is an example of code to generate a certificate request.

The form statement would look something like this:

```
<form name="UserCertForm" method=POST action="nextHTML" onSubmit="return makereq()">.
```

This code would need to be defined in the HTML before the JavaScript™.

```
<OBJECT classid="clsid:43F8F289-7A20-11D0-8F06-00C04FC295E1"
        CODEBASE="xenroll.dll"
        id=Enroll>
</OBJECT>
```

This is a JavaScript function that would be defined along with the HTML form that is used to collect the necessary data to create the certificate request.

```
function makereq() {
  var checkResult = "";
  var distNamePurpose = "";
  var distName = "";
  var certData = "";
  var errStr = "";

  // Still need to make sure that the fields are OK
  checkResult = validate(); // Function that will check the validity of the
                            // data, such as making sure required fields are
                            // filled in and that the state field is at least
                            // 3 chars, etc.
  if (checkResult == true) {
    // Create the distinguished name from the input fields
    distName  = "C="  + document.UserCertForm.countryregion.value;
    distName += ";ST=" + document.UserCertForm.stateprov.value;
    distName += ";L="  + document.UserCertForm.locality.value;
    distName += ";O="  + document.UserCertForm.orgname.value;
    distName += ";OU=" + document.UserCertForm.orgunitname.value;
    distName += ";CN=" + document.UserCertForm.commonname.value;

    Enroll.KeySpec = 1;
    Enroll.GenKeyFlags = 1;
    distNamePurpose = "1.3.6.1.4.1.311.2.1.21";
    certData = Enroll.CreatePKCS10(distName, distNamePurpose);

    if (certData == "") {
      // Certificate generation failed - put up an alert or something
      errStr = "The certificate request was not created";
      alert(errStr);
      return (false);
    }
    else {
      // Certificate generation OK - submit the request
      document.UserCertForm.certData.value = certData;
      return (true);
    }
  }
  else
    return (false);
}
```

API introduced: V5R2

Digital Certificate Management Exit Programs

If an exit program has been associated with an application, the exit program currently can be called for four different reasons:

- When the information about the exit program is updated.
- When the application is being deregistered.
- When the certificate associated with the application is updated or removed.
- When a Certificate Authority (CA) is added to or removed from the trust list for the application.

The information that is passed to the exit program is different based on the action that is being performed. The exit point format name that is passed to the exit program identifies the action that is being performed, and thus identifies what additional information is provided to the exit program.

The digital certificate management exit programs are:

- "Deregister Application for Certificate Use Exit Program" is called when an application that uses certificates is deregistered.
- "Register Application for Certificate Use Exit Program" on page 78 is called when the registration information for an application is changed using the Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API, the Add Exit Program (QUSADDEP, QusAddExitProgram) API, or the Add Exit Program (ADDEXITPGM) command.
- "Update Certificate Authority (CA) Trust Exit Program" on page 80 is called when a CA certificate is added to or removed from the list of trusted CA certificates for an application using Digital Certificate Manager (DCM).
- "Update Certificate Usage Exit Program" on page 82 is called when a certificate is updated for an application or removed from an application using Digital Certificate Manager (DCM).

Exit Programs

These are the Exit Programs for this category.

Deregister Application for Certificate Use Exit Program

Required Parameter Group:

1	Deregister application exit information	Input	Char(*)
2	Deregister indicator	Output	Char(1)

QSYSINC Member Name: ESYDRAPP
Exit Point Name: QIBM_QSY_CERT_APPS
Exit Point Format Name: DRAP0100

The Deregister Application for Certificate Use exit program is called when an application that uses certificates is deregistered using the Deregister Application for Certificate Use (QsyDeregisterAppForCertUse) API, the Remove Exit Program (QUSRMVEP, QusRemoveExitProgram) API, or the Remove Exit Program (RMVEXITPGM) command.

When an application is being deregistered, the user-written exit program associated with the registered application is called. The exit point supports an unlimited number of applications, but only one exit program for each application. (For information about registering an application for certificate use, see "Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API" on page 46.)

Note: The Deregister Application for Certificate Use exit point will not deregister the application if the user-written exit program indicates that the deregister operation is not allowed. If the exit program does not exist or cannot be called because of the multithreaded job action value, then the application will be deregistered.

Authorities and Locks

Authority to Exit Program Library
*EXECUTE

Authority to Exit Program
*USE

Required Parameter

Deregister application exit information
INPUT; CHAR(*)

Information needed by the exit program for notification of a deregister operation on the application. For details, see "Format of Deregister Application Exit Information."

Deregister indicator
OUTPUT; CHAR(1)

An indicator set by the exit program as to whether the deregister of the application is allowed. The possible values follow:

0 The application will not be deregistered.
1 The application will be deregistered.

Note: Any return value other than 1 will prevent the application from being deregistered.

Format of Deregister Application Exit Information

The following table shows the structure of the deregister application information for format DRAP0100. For a description of the fields in this format, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(20)	Exit point name
20	14	CHAR(8)	Exit point format name
28	1C	CHAR(100)	Application ID

Field Descriptions

Application ID. The ID of the application being deregistered.

Exit point format name. The format name for the Deregister Application for Certificate Use exit program. The possible format name is:

DRAP0100 The format name that is used when an application is being deregistered.

Exit point name. The name of the exit point that calls the exit program.

Exit program introduced: V4R4

Register Application for Certificate Use Exit Program

Required Parameter Group:

1	Register application exit information	Input	Char(*)
2	Register indicator	Output	Char(1)

QSYSINC Member Name: ESYRGAPP
Exit Point Name: QIBM_QSY_CERT_APPS
Exit Point Format Name: RGAP0100

The Register Application for Certificate Use exit program is called when the registration information for an application is changed using the Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API, the Add Exit Program (QUSADDEP, QusAddExitProgram) API, or the Add Exit Program (ADDEXITPGM) command.

When the information for a registered application is being changed, the user-written exit program associated with the registered application is called. The exit point supports an unlimited number of applications, but only one exit program for each application. (For information about registering an application that uses certificates, see "Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API" on page 46.)

Note: The Register Application For Certificate Use exit point does not change the application information if the user-written exit program indicates that the change operation is not allowed. If the exit program does not exist or cannot be called because of the multithreaded job action value, then the application information is changed.

Authorities and Locks

Authority to Exit Program Library
*EXECUTE

Authority to Exit Program
*USE

Required Parameter

Register application exit information
INPUT; CHAR(*)

Information needed by the exit program for notification of any changes to a registered application. For details, see "Format of Register Application Exit Information" on page 79.

Register indicator
OUTPUT; CHAR(1)

An indicator set by the exit program as to whether the change of the application information is allowed. The possible values follow:

0 The application information will not be changed.
1 The application information will be changed.

Format of Register Application Exit Information

The following table shows the structure of the register application information for format RGAP0100. For a description of the fields in this format, see "Field Descriptions."

Offset			
Dec	**Hex**	**Type**	**Field**
0	0	CHAR(20)	Exit point name
20	14	CHAR(8)	Exit point format name
28	1C	CHAR(100)	Application ID
128	80	CHAR(1)	Current client authentication required value
129	81	CHAR(1)	New client authentication required value
130	82	CHAR(1)	Current client authentication supported value
131	83	CHAR(1)	New client authentication supported value
132	84	CHAR(1)	Current limit CA certificates trusted value
133	85	CHAR(1)	New limit CA certificates trusted value

Field Descriptions

Application ID.

The ID of the application.

Current client authentication required value. The current value for the client authentication required indicator. The possible values follow:

0 Client authentication is not required.
1 Client authentication is required.

Current client authentication supported value. The current value for the client authentication supported indicator. The possible values follow:

0 Client authentication is not supported by this application.
1 Client authentication is supported by this application.

Current limit CA certificates trusted value. The current value for the limit Certificate Authority (CA) certificates trusted indicator. The possible values follow:

0 Application trusts all CA certificates that are trusted in the *SYSTEM certificate store.
1 Application trusts a subset of the CA certificates that are trusted in the *SYSTEM certificate store.

Exit point format name. The format name for the Register Application for Certificate Use exit program. The possible format name is:

RGAP0100 The format name that is used after application information is changed.

Exit point name. The name of the exit point that calls the exit program.

New client authentication required value. The new value for the client authentication required indicator. The possible values follow:

0 Client authentication is not required.
1 Client authentication is required.

New client authentication supported value. The new value for the client authentication supported indicator. The possible values follow:

0 Client authentication is not supported by this application.
1 Client authentication is supported by this application.

New limit CA certificates trusted value. The new value for the limit Certificate Authority (CA) certificates trusted indicator. The possible values follow:

0 Application trusts all CA certificates that are trusted in the *SYSTEM certificate store. If the current limit CA certificates trusted value is 1, then any CA certificates that are in the list of trusted CA certificates for the application will be removed.
1 Application trusts a subset of the CA certificates that are trusted in the *SYSTEM certificate store. If the current limit CA certificates trusted value is 0, then the application will not trust any of the CA certificates that are trusted in the *SYSTEM certificate store until they are added to the list of trusted CA certificates for the application using Digital Certificate Manager (DCM).

Note: The Update Certificate Authority (CA) Trust exit program will not be called for the CA certificates that are removed from the list of trusted CA certificates for the application because of a change to this value.

Exit program introduced: V5R1

Update Certificate Authority (CA) Trust Exit Program

Required Parameter Group:

1	Update Certificate Authority (CA) trust exit information	Input	Char(*)

QSYSINC Member Name: ESYUPDCA
Exit Point Name: QIBM_QSY_CERT_APPS
Exit Point Format Name: CATR0100

The Update Certificate Authority (CA) Trust exit program is called when a CA certificate is added to or removed from the list of trusted CA certificates for an application using Digital Certificate Manager (DCM).

When the trust status of a CA certificate for an application is changed, the user-written exit program associated with the registered application is called. The exit point supports an unlimited number of applications, but only one exit program for each application. (For information about registering an application that uses certificates, see "Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API" on page 46.)

Note: The Update Certificate Authority (CA) Trust exit program is not be called if the Limit CA certificates trusted indicator for the application is set to 0 (the application trusts all CA certificates that are trusted in the *SYSTEM certificate store) and the trust status for one of the CA certificates in the *SYSTEM certificate store is changed.

Note: The Update Certificate Authority (CA) Trust exit program ignores any return codes or error messages that are sent from the exit program.

Authorities and Locks

Authority to Exit Program Library
*EXECUTE

Authority to Exit Program
*USE

Required Parameter

Update Certificate Authority (CA) trust exit information
INPUT; CHAR(*)

Information needed by the exit program for notification of any CA certificate trust changes for the application. For details, see "Format of Update Certificate Authority (CA) Trust Exit Information."

Format of Update Certificate Authority (CA) Trust Exit Information

The following table shows the structure of the update CA trust information for format CATR0100. For a description of the fields in this format, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(20)	Exit point name
20	14	CHAR(8)	Exit point format name
28	1C	CHAR(100)	Application ID
128	80	CHAR(1)	Action
129	81	CHAR(1)	Trusted CA certificate ID type
130	82	CHAR(2)	Reserved
132	84	BINARY(4)	Offset to trusted CA certificate ID
136	88	BINARY(4)	Length of trusted CA certificate ID
		CHAR(*)	Trusted CA certificate ID

Field Descriptions

Action.

The action being performed on the trusted CA certificate. The possible values follow:

0 The trusted CA certificate is being added to the list of trusted CA certificates for the application.
1 The trusted CA certificate is being removed from the list of trusted CA certificates for the application.

Application ID. The ID of the application.

Trusted CA certificate ID. The ID for the trusted CA certificate being added or removed.

Trusted CA certificate ID type. The type of the trusted CA certificate ID. The possible value follows:

1 The trusted CA certificate ID is the label for the certificate.

Exit point format name. The format name for the Update Certificate Authority (CA) trust exit program. The possible format name is:

CATR0100 The format name that is used after a CA certificate is added or removed from the trust list for an application.

Exit point name. The name of the exit point that calls the exit program.

Length of trusted CA certificate ID. The length of the trusted CA certificate ID.

Offset to trusted CA certificate ID. The offset to the start of the trusted CA certificate ID.

Reserved. An ignored field.

Exit program introduced: V5R1

Top | Security APIs | APIs by category

Update Certificate Usage Exit Program

Required Parameter:

1	Update certificate usage exit information	Input	Char(*)

```
QSYSINC Member Name:  ESYUPDCU
Exit Point Name:  QIBM_QSY_CERT_APPS
Exit Point Format Name:  CERT0100
```

The Update Certificate Usage exit program is called when a certificate is updated for an application or removed from an application using Digital Certificate Manager (DCM).

When a certificate for an application is changed, the user-written exit program associated with the registered application is called. The exit point supports an unlimited number of applications, but only one exit program for each application. (For information about registering an application that uses certificates, see "Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API" on page 46.)

Note: The Update Certificate Usage exit point ignores any return codes or error messages that are sent from the exit program.

Authorities and Locks

Authority to Exit Program Library
*EXECUTE

Authority to Exit Program
*USE

Required Parameter

Update certificate usage exit information
INPUT; CHAR(*)

Information needed by the exit program for notification of any certificate changes for the application. For details, see "Format of Update Certificate Usage Exit Information" on page 83.

Format of Update Certificate Usage Exit Information

The following table shows the structure of the update certificate usage information for format CERT0100. For a description of the fields in this format, see "Field Descriptions."

Offset		Type	Field
Dec	Hex		
0	0	CHAR(20)	Exit point name
20	14	CHAR(8)	Exit point format name
28	1C	CHAR(100)	Application ID
128	80	CHAR(1)	Action
129	81	CHAR(1)	Certificate ID type
130	82	CHAR(2)	Reserved
132	84	BINARY(4)	Offset to certificate store
136	88	BINARY(4)	Length of certificate store
140	8C	BINARY(4)	Offset to certificate ID
144	90	BINARY(4)	Length of certificate ID
		CHAR(*)	Certificate store
		CHAR(*)	Certificate ID

Field Descriptions

Action. The action being performed on the certificate. The possible values follow:

0 The certificate is being added to the application.
1 The certificate is being changed for the application.
2 The certificate is being removed from the application.

Application ID. The ID of the application.

Certificate ID. The ID for the updated certificate.

Certificate ID type. The type of the certificate ID. The possible value follows:

1 A certificate ID is the label for the certificate.

Certificate store. The path name where the certificate is stored. The path name will be specified in the coded character set ID (CCSID) of the job. The following special value may be specified:

**SYSTEM* The certificate is stored in the system certificate store.

Exit point format name. The format name for the Update Certificate Usage exit program. The possible format name is:

CERT0100 The format name that is used after a certificate is updated for an application.

Exit point name. The name of the exit point that calls the exit program.

Length of certificate ID. The length of the certificate ID.

Length of certificate store. The length of the certificate store.

Offset to certificate ID. The offset to the start of the certificate ID.

Offset to certificate store. The offset to the start of the certificate store.

Reserved. An ignored field.

Exit program introduced: V4R4

Top | Security APIs | APIs by category

Code license and disclaimer information

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

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

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