Security APIs (V5R2)

Digital Certificate Management APIs

Table of 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 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) associates a certificate with an OS/400 user profile.
- Add Validation List Certificate (QSYADDVC,QsyAddVldlCertificate) adds a certificate to a validation list.
- Check Validation List Certificate (QSYCHKVC, QsyCheckVldlCertificate) determines whether a certificate is in a validation list.
- Deregister Application for Certificate Use (QsyDeregisterAppForCertUse) removes an application and all associated certificate information from the registration facility.
- Find Certificate User (QSYFNDCU, QsyFindCertificateUser) finds the user that is associated with a certificate.
- ≫Generate and Sign User Certificate Request (QYCUGSUC) generates a user certificate request and then signs the certificate request using the local Certificate Authority (CA).≪
- List User Certificates (QSYLSTUC, QsyListUserCertificates) lists the certificates in the user profile.
- List Validation List Certificates (QSYLSTVC, QsyListVldlCertificates) lists the certificates in the validation list.
- Open List of User Certificates (QSYOLUC) provides a list of user certificates associated with a user.
- Parse Certificate (QSYPARSC, QsyParseCertificate) parses a certificate and puts the results in the caller's storage.
- Register Application for Certificate Use (QsyRegisterAppForCertUse) registers an application with the registration facility.
- Remove User Certificate (QSYRMVUC, QsyRemoveUserCertificate) removes a certificate from an OS/400 user profile.
- Remove Validation List Certificate (QSYRMVVC, QsyRemoveVldlCertificate) removes a certificate from a validation list.
- ≫Sign User Certificate Request (QYCUSUC) signs a user certificate request using the local Certificate Authority (CA).≪

Add User Certificate (QSYADDUC, QsyAddUserCertificate) API

```
Required Parameter Group:

    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

Service Program: QSYDIGID

Threadsafe: Yes
```

The Add User Certificate (OPM, QSYADDUC; ILE, QsyAddUserCertificate) API associates a certificate with an OS/400 user profile.

A common scenario is that only one certificate is associated with an OS/400 user profile at any given time, but more than one certificate may be associated with the same OS/400 user profile if each certificate is unique. A reason for having more than one certificate associated with an OS/400 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 OS/400 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 OS/400 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 OS/400 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, Public Key Cryptography Standard 6 (PKCS-6) format.

3   Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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.
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.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V4R2

Add Validation List Certificate (QSYADDVC, QsyAddVldlCertificate) API

```
Required Parameter Group:

   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

Service Program: QSYDIGID

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, Public Key Cryptography Standard 6 (PKCS-6) format.

      3   Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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:

Check Validation List Certificate (QSYCHKVC,QsyCheckVldlCertificate) AP

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, Public Key Cryptography Standard 6 (PKCS-6) format.

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, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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.
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

Deregister Application for Certificate Use (QSYDRGAP, QsyDeregisterAppForCertUse) API

```
Required Parameter Group:

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


Service Program Name: 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

Find Certificate User (QSYFNDCU,QsyFindCertificateUser) API

Required Parameter Group:

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

Service Program: QSYDIGID

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, Public Key Cryptography Standard 6 (PKCS-6) format.

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, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V4R2

»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.

```
%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

List User Certificates (QSYLSTUC, QsyListUserCertificates) API

Required Parameter Group:

1	Qualified user space name	Input	Char(20)
2	User name	Input	Char(10)
3	Format name	Input	Char(8)
4	Selection control	Input	Char(*)
5	Error code	I/O	Char(*)

Default Public Authority: *USE

Service Program: QSYDIGID

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 OS/400 user profile at any given time, but more than one certificate may be associated with the same OS/400 user profile if each certificate is unique. The same certificate is not allowed to be associated with more than one OS/400 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 OS/400 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, 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 OS/400 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
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(10)

The name of the user profile.

The following is a valid selection:

*CURRENT   The user profile that is currently running.

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:

[*CERT0100*](#)   Certificates in ASN.1 format

[*CERT0200*](#)   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](#).

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 Space Format for List APIs. For details about the formats in the list data section, see Certificate Format CERT0100 (ASN.1) and Certificate Format CERT0200 (Plain Text).

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

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
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

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
		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
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
		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)	Returned length of this certificate and format information
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.

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

Certificate information fields. The actual data i the certificate. Specific fields can be accessed by using the offset to that 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 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. The length indicates bytes of hexadecimal numbers.

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. This length refers to a field of hexadecimal bytes.

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. If the length is longer than 8, the characters after the date represent time of day according to the type used by the certificate authority when the certificate was created.

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. If the length is longer than 8, the characters after the date represent time of day according to the type used by the certificate authority when the certificate was created.

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 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. This offset refers to a field of hexadecimal bytes.

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. This offset refers to a field of hexadecimal bytes.

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.

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.

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 in 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.
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.
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.
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

List Validation List Certificates (QSYLSTVC,QsyListVldlCertificates) API

```
Required Parameter Group:


    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

Service Program: QSYDIGID

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:

      CERT0100   Certificates in Abstract Syntax Notation 1 (ASN.1) format

      CERT0200   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.

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:

- 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 Space Format for List APIs.

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

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

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.

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.
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.
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, see Format of Receiver Variable.

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](#) 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:

CERT0100   Certificates in ASN.1 format in ASCII encoding.

CERT0200   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 Receiver Variable

The following tables describe the order and format of the data returned in the receiver variable. For detailed descriptions of the fields in the tables, see [Field Descriptions](#).

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			
Dec	Hex	Type	Field

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
		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. 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			
Dec	Hex	Type	Field
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
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
		ARRAY(*) of CHAR	Certificate information fields

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 fields. 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 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 serial number. The length of the field that indicates the serial number. The length indicates bytes of hexadecimal numbers.

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. This length refers to a field of hexadecimal bytes.

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. If the length is longer than 8, the characters after the date represent time of day

according to the type used by the certificate authority when the certificate was created.

Length of validity period end. The length of the field that indicates the ending date of the validity period. 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. If the length is longer than 8, the characters after the date represent time of day according to the type used by the certificate authority when the certificate was created.

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 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 serial number. The offset to the field that indicates the serial number. This offset refers to a field of hexadecimal bytes.

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. This offset refers to a field of hexadecimal bytes.

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.

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.

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.
CPF9872 E	Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V4R3

Parse Certificate (QSYPARSC, QsyParseCertificate) API

Required Parameter Group:

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

Service Program: QSYDIGID

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, Public Key Cryptography Standard 6 (PKCS-6) format.

    3   Base 64 encoded version of the entire X.509 public key certificate in ASN.1 DER encoding, Public Key Cryptography Standard 6 (PKCS-6) format. 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:

    CERT0200   All text fields available.

    CERT0210   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.

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).

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.

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
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	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 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. The length indicates bytes of hexadecimal numbers.

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. This length refers to a field of hexadecimal bytes.

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. If the length is longer than 8, the characters after the date represent time of day according to the type used by the certificate authority when the certificate was created.

Length of validity period end. The length of the field that indicates the ending date of the validity period. 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. If the length is longer than 8, the characters after the date represent time of day according to the type used by the certificate authority when the certificate was created.

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 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. This offset refers to a field of hexadecimal bytes.

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. This offset refers to a field of hexadecimal bytes.

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.
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.
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

Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API

Required Parameter Group:

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

Service Program Name: 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

*SECADM Special Authority

> A user profile name for the application user profile control key is specified.

Registration Lock

> *EXCL

Required Parameter Group

Application ID

> INPUT; CHAR(*)

> The application ID to register. IBM-supplied iSeries applicationss 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_OS400_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.

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 and the application is associated with a certificate in the *SYSTEM certificate store, 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. Refer to Digital Certificate Management exit programs 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. Refer to Qualified Message File Format 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 function 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.

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

Remove User Certificate (QSYRMVUC, QsyRemoveUserCertificate) API

```
Required Parameter Group:

    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

Service Program: QSYDIGID

Threadsafe: Yes
```

The Remove User Certificate (OPM, QSYRMVUC; ILE, QsyRemoveUserCertificate) API removes a certificate from an OS/400 user profile.

Authorities and Locks

User Profile Authority

> If the user profile specified is not the current user for the job, 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 holds 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 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, Public Key Cryptography Standard 6 (PKCS-6) format.

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, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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.
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.

CPF9872 E Program or service program &1 in library &2 ended. Reason code &3.

API Introduced: V4R2

Remove Validation List Certificate (QSYRMVVC, QsyRemoveVldlCertificate) API

```
Required Parameter Group:

    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

Service Program: QSYDIGID

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, Public Key Cryptography Standard 6 (PKCS-6) format.

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, Public Key Cryptography Standard 6 (PKCS-6) format. 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.
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

»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

See Code disclaimer information for information pertaining to code examples.

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 = "CN=" + document.UserCertForm.commonname.value;
    distName += ";OU=" + document.UserCertForm.orgunitname.value;
    distName += ";O="  + document.UserCertForm.orgname.value;
    distName += ";L="  + document.UserCertForm.locality.value;
    distName += ";ST=" + document.UserCertForm.stateprov.value;
    distName += ";C="  + document.UserCertForm.countryregion.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);
}
```

«
_____

_____

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](#) is called when an application that uses certificates is deregistered.
- [Register Application for Certificate Use](#) 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](#) 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](#) is called when a certificate is updated for an application or removed from an application using Digital Certificate Manager (DCM).

Deregister Application for Certificate Use Exit Program

```
Required Parameter Group:

  1    Deregister application exit      Input       Char(*)
       information
  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 the Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API.

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       Input      Char(*)
       information
  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 the [Register Application for Certificate Use](#) (QSYRGAP, QsyRegisterAppForCertUse) API.

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.

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		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)	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   Input        Char(*)
       (CA) trust exit information

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 the Register Application for Certificate Use QSYRGAP, QsyRegisterAppForCertUse) API.

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

Update Certificate Usage Exit Program

```
Required Parameter:

  1   Update certificate usage     Input        Char(*)
      exit information



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 the Register Application for Certificate Use (QSYRGAP, QsyRegisterAppForCertUse) API.

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.

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