

## 3 CL Commands to Manage Digital Certificates

[System\*i\*NEWS Magazine](#)

[Carsten Flensburg](#)

Carsten Flensburg

Wed, 03/28/2012 (All day)

3 CL commands help you manage business-critical IBM i digital certificates



I recently enjoyed spending a couple of days at a IBM technical convention here in Copenhagen. The many interesting and current topics discussed at the event included digital certificates and their vital role in authenticating and securing a modern communication and transaction infrastructure.

One of the issues that came up during the discussion at the IBM convention was a problem relating to certificates expiring without the system issuing any notice of this. Using the IBM i's Digital Certificate Manager (DCM), you can list and check certificates approaching expiration within a specified number of days. (For more information about using DCM, see the sidebar "How to Start the Digital Certificate Manager," after the main article.) But currently there's no system function that automatically issues certificate-expiration notification messages to the QSYSOPR message queue, or similar system message queues, where critical notifications relating to other system components would normally surface.

However, IBM supplies a digital certificate management API that provides program access to the functions of listing and expiration-checking certificates. The Retrieve Certificate Information (QycuRetrieveCertificateInfo) ILE API and its Original Program Model (OPM) counterpart were both added by IBM in release 6.1. In this article, I'll provide an overview of the Retrieve Certificate Information API along with several CL commands I created based on the API that can make it easier for you to manage your IBM i digital certificates and track their expiration dates. (See the sidebar "Creating the CHKCERTEXP, WRKCERT, and DSPCERT Commands" for instructions on downloading and creating the CL commands discussed in this article.)

### Digital Certificates on IBM i

A digital certificate is an electronic credential that you can use to establish proof of identity in an electronic transaction—for example, a transaction performed on the Internet—when you transfer sensitive personal information or close a purchase by submitting an electronic payment using your credit card. On the IBM i, the DCM browser-based user interface is provided to enable users to create, import, and configure certificates for a variety of purposes and to perform many other certificate-related tasks.

Digital certificates are integral to the process of SSL (aka Transport Layer Security -TLS) performing authentication services and encrypting transaction data as part of IP-based protocols such as HTTP, FTP, and Telnet. Digital certificates also provide the encryption backbone for communication protocols such as Virtual Private Network (VPN) and, for example, object signing, which allows you to verify both the integrity of an object's contents and its source of origin, regardless of whether the object is in transit across the Internet or stored on a remote system.

In regard to certificate expiration, because it often takes some time to complete a certificate-renewal process involving an external certificate issuer, being unaware of an expiring server certificate on your company's website could impede your website from performing secure transactions until the certificate has been renewed. Fortunately, the Retrieve Certificate Information API offers help. Let's take a closer look at the API, then move into a discussion of the certificate-management commands I created that use the API.

### Noteworthy API Parameters

When I perused the Retrieve Certificate Information API documentation in IBM's Information Center, it seemed straightforward. Figure 1 shows the QycuRetrieveCertificateInfo API's parameter list.

Figure 1: QycuRetrieveCertificateInfo API parameter list

## Required Parameter Group:

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

Rather than explain each parameter in detail, I'll mention a few of the options offered by the Retrieve Certificate Information API and considerations involved in deciding which of these options to employ.

The *Certificate store name* parameter can be specified as a simple string in the Coded Character Set Identifier (CCSID) currently in effect for the job calling the API. This format is called OBJNo100. Alternatively you can use an API path name structure, supporting a number of parameters that define to the API how to interpret the path name submitted. You specify a CCSID, a country or region ID, a language ID, and the path type as either a character string or a pointer to a character string, the length of path name, and the path name delimiter character.

Most of the configuration parameters in the API path name structure take a default value pointing to the corresponding job attribute currently in effect, and if chosen, the default values in turn provide an end effect similar to that achieved by the OBJNo100 format. The API path name format is called OBJNo200 and is especially useful when the path name's CCSID is not known in advance, a special delimiter is used, or the path name is referenced by a pointer value.

When the Retrieve Certificate Information API was initially introduced, the *Certificate store password* parameter only had one option: You had to specify the actual certificate store password to gain access to the certificate information within. As a result of a recent PTF, yet another option exists. You are now allowed to specify the special value \*NOPWD, causing the API to retrieve the actual certificate password from the system internal stash password index. This, of course, somewhat reduces the protection imposed by the store password in the first place.

However, the API requires \*ALLOBJ and \*SECADM special authority from the user profile calling it, so some safeguard remains, at least if you're cautious about which user profiles are allotted that level of special authority. The upside is the ability to have a program retrieve certificate store information, without having to store the password somewhere for the program to be able to access it-which would compromise security.

The final API parameter I will mention is the *Selection control* parameter, which enables you to apply three different selection criteria in retrieving the list of certificates contained in the certificate store. The following criteria are supported:

- Certificate type: Certificate Authority (CA) or server/client
- Expiration days: the number of days within which the certificate is due to expire
- Certificate label: the label identifying the certificate

Certificate type and expiration days can be combined, whereas certificate label excludes specification of any other criterion. As you can see, the expiration days criterion does the hard work for you when trying to locate certificates approaching expiration.

Note that the QycuRetrieveCertificateInfo ILE API is found in the service program QYCUCERTI, which is placed in library QICSS, the product library of the DCM licensed program. The compiler will therefore not be able to automatically pick up the API in question, but instead relies on you to specify the service program explicitly when creating the program referencing it. Alternatively, you must add the aforementioned service program to a binding directory named in the header specification of the module calling the API.

## Bumps in the Road

Based on my research, I felt rather confident when I first tried to call the Qy cuRetrieveCertificateInfo API. Much to my surprise, however, all my attempts to call the API failed miserably. I tried all the tricks of the trade in adjusting the API prototype as well as the parameters submitted, all to no avail. Eventually I had to call on IBM support and open a problem management record (PMR) in my pursuit of success in using the API.

Luckily, it turned out that a PMR had already been opened because of issues similar to the ones I had run into and that a PTF was in the works at this point. I was allowed to download the PTF in its test version; once I loaded and applied the PTF, I no longer had any problems calling the Qy cuRetrieveCertificateInfo API. You can find the relevant APAR and PTF IDs in the sidebar "Learn More About Using Digital Certificates on IBM i."

### The Check Certificate Expiration Command

After I had resolved the issue with calling the Qy cuRetrieveCertificateInfo API, I was able to move on to creating the Check Certificate Expiration (CHKCERTEXP) CL command to assist in digital certificate management on the IBM i. Creating the CHKCERTEXP CL command was, to some extent, as simple as exposing the relevant API parameters in the command interface, as shown in Figure 2.

Figure 2: Check Certificate Expiration (CHKCERTEXP) command prompt

```
-----
                                Check Certificate Expiration (CHKCERTEXP)

Type choices, press Enter.

Certificate store . . . . . *SYSTEM
Certificate type . . . . . *ALL          *ALL, *SERVER, *CA
Expiration days . . . . . 60            1-365
Include expired certificates . . *YES      *YES, *NO
Certificate store password . . . *NOPWD
Message queue . . . . . *USRPRF      Name, *USRPRF, *SYSOPR...
    Library . . . . . *LIBL          Name, *LIBL, *CURLIB
                                + for more values
                                *LIBL
```

The CHKCERTEXP command has an associated help text panel group that explains the command and all its parameters. The CHKCERTEXP command uses the Send Nonprogram Message (QMHSNDM) API to send a predefined message description for each expired certificate returned by the Qy cuRetrieveCertificateInfo API.

Once I had finished writing the command's code, I tested CHKCERTEXP by running the following command:

```
CHKCERTEXP CERTSTORE(*SYSTEM)
  TYPE(*ALL)
  DAYS(60)
  INCLUDE(*NO)
  STOREPWD(*NOPWD)
  MSGQ(*SYSOPR)
```

After this command finished executing, I checked the QSYSOPR message queue and found the warning notification message shown in Figure 3a.

Figure 3A: Certificate-expiration warning message in the QSYSOPR message queue

```
Display Messages

                                System:  WYNDHAMW
Queue . . . . . :  QSYSOPR          Program . . . . :  *DSPMSG
  Library . . . . :  QSYS            Library . . . . :
Severity . . . . :  99              Delivery . . . . :  *HOLD

Type reply (if required), press Enter.
Certificate expiration warning. Certificate expires on 06-01-12 20:17:31.
```

I prompted the command by pressing the F1 key with the cursor placed on the message text to see the complete message, including the second-level message text and the additional message information shown in Figure 3b.

Figure 3B: Additional message information

```
Additional Message Information

Message ID . . . . . : CBX1001      Severity . . . . . : 00
Message type . . . . . : Information
Date sent . . . . . : 18-12-11      Time sent . . . . . : 17:59:06

Message . . . . . : Certificate expiration warning. Certificate expires on
06-01-12 20:17:31.
The certificate labeled
'b97d04f8292c93c28fed3cfd8b13c1f8b_dcd82ba6-0b68-4034-b232-2abc22b8274d' in
the *SYSTEM certificate store expires on 06-01-12 20:17:31. The subject
common name associated with the certificate is '1013694731-5'.
```

To add a job schedule entry ensuring that a check against expiring certificates is performed every Monday morning, you would run the following Add Job Schedule Entry (ADDJOBSCDE) command:

```
ADDJOBSCDE JOB(CHKCERTEXP) CMD(CHKCERTEXP DAYS(60) INCLUDE(*NO) MSGQ(*SYSOPR))
FRQ(*WEEKLY)
SCDDATE(*NONE)
SCDTIME(070000)
JOB(*USRPRF)
JOBQ(*JOBQ)
USER(QSECOFR)
```

Make sure you adapt the ADDJOBSCDE command's JOBQ, JOBQ, and USER parameters to reflect your preferred job runtime environment. Note that special authorities \*ALLOBJ and \*SECADM are required for the user profile running the CHKCERTEXP command.

Two More Certificate Commands

After I had finished the CHKCERTEXP command, I decided to also write a Work with Certificates (WRKCERT) command. Although DCM offers access to the type of certificate information and functionality implemented by the WRKCERT command, the WRKCERT command does not require you to start the HTTP server's \*ADMIN instance in order to list and display certificates. So for some situations you might find it quite handy to have a green-screen command that gives you instant access to current certificate information.

The WRKCERT command prompt, shown in Figure 4, is similar to the CHKCERTEXP command prompt.

Figure 4: Work with Certificates (WRKCERT) command prompt

```
Work with Certificates (WRKCERT)

Type choices, press Enter.

Certificate store . . . . . *SYSTEM
Certificate type . . . . . *ALL      *ALL, *SERVER, *CA
Expiration days . . . . . *NONE     1-365, *NONE
Include expired certificates . . *YES  *YES, *NO
Certificate store password . . . *NOPWD
Output . . . . . *      *, *PRINT
Certificate store path . . . . . *NONE
```

Instead of sending messages, however, the WRKCERT command lists, either by displaying or printing, the certificates meeting the specified selection criteria. Figure 5 shows an example of how the resulting list looks on a display panel. The available list options enable you to display or print the certificate details, as well as display any key usage extensions in effect for the selected certificate.

Figure 5: Work with Certificates list panel

```
Work with Certificates                                WYNDHAMW                                18-12-11 18:04:43

Certificate store . : *SYSTEM

Type options, press Enter.
5=Display certificate  6=Print certificate  8=Display usage extensions
```

```

Certificate
Opt  Label
VeriSign Class 3 Extended Validation SSL CA
VeriSign Class 3 Public Primary Certification Authority - G5
VeriSign Class 3 Secure Server CA - G2
OU=VeriSign Trust Network,OU=(c) 1998 VeriSign, Inc. - For authorized use
Class 3 Public Primary Certification Authority EV
VeriSign Trust Network
Verisign Class 3 Public Primary Certification Authority
Thawte Personal Premium CA
Thawte Personal Freemail CA

More...

Parameters or command
==>
F3=Exit    F4=Prompt    F10=Check certificate expiration  F11=Validity period
F17=Top    F18=Bottom   F22=Display entire field value   F24=More keys

```

Key usage extensions define the purpose of the public key contained in a certificate. You can use them to restrict the public key to as few or as many operations as needed. For example, if you have a key used only for signing or verifying a signature, only the digital signature and/or non-repudiation extensions should be enabled. For a complete list of key usage extensions, see the link in "Learn More About Using Digital Certificates on IBM i."

The *Display certificate* and *Print certificate* list options are both based on the Display Certificate (DSPCERT) command, which I created specifically for the purpose of displaying certificate information. In addition to some of the parameters you recognize from the CHKCERTEXP and WRKCERT commands, the DSPCERT command's primary parameter is the *Certificate label* 4, which identifies the certificate within the specified certificate store that you want to display or print. Figure 6 shows the DSPCERT command prompt.

Figure 6: Display Certificate (DSPCERT) command prompt

```

Display Certificate (DSPCERT)

Type choices, press Enter.

Certificate label . . . . .

Certificate store . . . . . *SYSTEM
Certificate store password . . . *NOPWD
Output . . . . . *          *, *PRINT
Certificate store path . . . . . *NONE

```

Figure 7 shows an example of how the resulting Display Certificate display panel looks.

Figure 7: Display Certificate panel

```

Display Certificate                                WYNDHAMW                                18-12-11  18:07:30

Certificate label . . . :  VeriSign Class 3 Extended Validation SSL CA

Additional information:
Key size . . . . . :  2048
Private key . . . . . :  *NO
Private key label . . . :

Serial number . . . . . :  5B7759C61784E15EC727C0329529286B

Validity start . . . . . :  08-11-06  01:00:00
Validity end . . . . . :  08-11-16  00:59:59
Trusted . . . . . :  *YES
Storage location . . . :  *SEW
Key usage extension . . :  *YES
Usage extensions . . . :  KeyCertSign CRLSign

More...

F3=Exit    F5=Refresh    F12=Cancel

```

More Control Over Digital Certificates

The CHKCERTEXP, WRKCERT, and DSPCERT commands can help you stay on top of digital certificates on your IBM i, an essential part of doing business online. I hope you find the commands helpful-let me know how they work for you.

**Carsten Flensburg** is a *System iNEWS* senior technical editor and has been an IBM i programmer since 1992. His focus areas are modular system design, API programming, system integration, and communication. Carsten currently works as an IBM i application development manager for a European vacation rental company called Novasol, which is a part of the U.S.-based Wyndham Worldwide Corporation. Carsten's first experiences with *System iNEWS* date back to the early 1990s, when he became a fan of Paul Conte's thought-provoking articles. Carsten lives in Copenhagen, Denmark, with his wife, Dorte, and his two children, Julian and Emilie.

Sidebar 1: How to Start the IBM i's Digital Certificate Manager

Perform the following tasks to start the IBM i's Digital Certificate Manager (DCM).

- 1. Install DCM: Release 7.1 - Product 5770SS1 (OS) Option 34.
- 2. Install IBM HTTP Server for i: Release 7.1 - Product 5770DG1.
- 3. Use System i Navigator to start the HTTP Server Administrative server, as follows:
  - a. In System i Navigator expand your system > Network > Servers > TCP/IP.
  - b. Right-click HTTP Administration.
  - c. Select Start.
  - d. Alternatively, from a command line issue the following command:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

- 4. Open a web browser and enter `http://your_system_name:2001` to load the IBM Systems Director Navigator for i5/OS web console.
- 5. From the welcome page click the i5/OS Tasks Page link.
- 6. Select Digital Certificate Manager from the list of products on the i5/OS Tasks page to access the DCM user interface.
- 7. The left frame of Digital Certificate Manager (DCM) is the task navigation frame. You can use this frame to select a wide variety of tasks for managing certificates as well as the applications that use them.

You can use the Display Software Resource (DSPSRWRSC) command to verify that you have the required licensed program products and options installed. Figure A shows the information about DCM and IBM HTTP Server for i displayed on a release 7.1 system after you run the DSPSRWRSC command.

Figure A: DSPSRWRSC command information

Resource ID	Option	Feature	Feature Type	Library	Release	Description
5770SS1	34	5050	*CODE	QICSS	V7R1M0	Digital Certificate Manager
5770SS1	34	2924	*LNG	QICSS	V7R1M0	Digital Certificate Manager
5770DG1	*BASE	5050	*CODE	QHTTPSVR	V7R1M0	IBM HTTP Server for i
5770DG1	*BASE	2924	*LNG	QHTTPSVR	V7R1M0	IBM HTTP Server for i

For more information about DCM and how to manage certificates using this tool, see the links listed in the sidebar ""Learn More About Using Digital Certificates on IBM i."

Sidebar 2: Creating the CHKCERTEXP, WRKCERT, and DSPCERT Commands

You can download the code for the three commands discussed in the main article—Check Certificate Expiration (CHKCERTEXP), Work with Certificates (WRKCERT), and Display Certificate (DSPCERT)—by going to SystemiNetwork.com and clicking the code tab at the top right on the page.

Figure A lists the resources involved in creating the CHKCERTEXP, WRKCERT, and DSPCERT commands.

**Figure A: Digital certificate CL command resources**

Filename	Object Type	Associated CL Command	Description
CBX608	RPGLE	WRKCERT	CPP
CBX608E	RPGLE	WRKCERT	UIM exit program
CBX608H	PNLGRP	WRKCERT	Help
CBX608P	PNLGRP	WRKCERT	Panel group
CBX608V	RPGLE	WRKCERT	VCP
CBX609	RPGLE	DSPCERT	CPP
CBX609H	PNLGRP	DSPCERT	Help
CBX609P	PNLGRP	DSPCERT	Panel group
CBX609V	RPGLE	DSPCERT	VCP
CBX609X	CMD	DSPCERT	---
CBX610	RPGLE	CHKCERTEXP	CPP
CBX610H	PNLGRP	CHKCERTEXP	Help
CBX610V	RPGLE	CHKCERTEXP	VCP
CBX610X	CMD	CHKCERTEXP	---
CBX608M	CLP	WRKCERT	Build command
CBX609M	CLP	DSPCERT	Build command
CBX610M	CLP	CHKCERTEXP	Build command

To create these objects, compile and run the CBX608M, CBX609M, and CBX610M CL programs, in that sequence, following the instructions in the source header. You'll also find compilation instructions in the respective source headers.

### Sidebar 3: Learn More About Using Digital Certificates on IBM i

Recent APARs and corresponding PTFs required for successfully running the digital certificate CL commands:

- [APAR SE48408 6.1](#)
- [APAR SE48414 7.1](#)
- [PTF SI43880 6.1 1000](#)
- [PTF SI43881 7.1 1000](#)

IBM API documentation:

- [Retrieve Certificate Information \(QYCURLVCL, QycuRetrieveCertificateInfo\) API](#)
- [Digital Certificate Manager](#)
- [Configuring DCM](#)
- [Send Nonprogram Message](#)
- [API Path Name Format](#)
- [API Error Code Parameter Format](#)

Additional documentation:

- [Digital certificate article](#)
- [Key usage extensions](#)
- [.kdb file extension](#)

**Source URL:** <http://iprodeveloper.com/security/3-cl-commands-manage-digital-certificates>