

APIs by Example: IBM PTFs New Retrieve IOA Cache Battery Information API

[System i Network Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 01/05/2012 - 12:12pm

IBM normally introduces new APIs as part of a release upgrade, but at times IBM also employs PTFs to add new APIs between releases. In the previous installment of APIs by Example, I presented a couple of new Zip APIs ([the QzipZip and QzipUnzip APIs](#)), which had recently been added to release 7.1 by means of a PTF. Today I discuss another PTF'd addition to the IBM API inventory. As opposed to the Zip APIs that were only made available to release 7.1, the new Retrieve IOA Cache Battery Information (QSMRTVBTC) API is offered for the three most recent releases. So even if you're still on release 5.4, you'll be able to download and install the API, should you feel inclined to do so.

In today's article I use the new QSMRTVBTC API as the backbone for three new CL commands. I provide examples of how to exploit the API in terms of building the code around it, as well as supporting your operations staff by giving them immediate access to information otherwise involving the Start Service Tools (STRSST) command and employing either the Hardware Service Manager or the Display/Alter/Dump services. IBM also recently released another PTF allowing you to display or print IOA cache battery information by calling the QSMRTTCC program delivered by the PTF. I, however, prefer to be in charge of the coding myself, because that lets me design the API interfaces to reflect my personal preferences and requirements, and that's how the Work with IOA Cache Battery Status (WRKCCBSTS), the Display IOA Cache Battery Data (DSPCCBDTA), and the Check IOA Cache Battery Status (CHKCCBSTS) CL commands came about.

Because the QSMRTVBTC API was added by means of a PTF, the online API documentation in the IBM Information Center has not yet been updated to reflect the addition of the new API. All the QSMRTVBTC API information that will eventually appear in the IBM Information Center was, however, included in the cover letter of the PTFs installing the API, as well as the associated APARs. I've added links to both API cover letters and APARs at the end of this article; but to save you the trouble at this point, here's what the QSMRTVBTC API's parameter list looks like:

Retrieve IOA Cache Battery Information (QSMRTVBTC) API

Required Parameter Group:

1	Receiver variable	Output	Char(*)
2	Length of receiver variable	Input	Binary(4)
3	Format name	Input	Char(8)
4	Time zone	Input	Char(10)
5	Error code	I/O	Char(*)

Default public authority: *USE

The *Receiver variable* parameter is where the QSMRTVBTC API will put the cache battery information returned by the API, up to the length specified by the second parameter. As of release 6.1, ILE RPG supports data structures and character variables up to lengths of 16773104 bytes, whereas these lengths for earlier releases were limited to 65535 bytes. So in order to provide support for a return variable length larger than 65535 bytes in the code I present today also at release 5.4, I manually allocate the storage required by the API.

In case the initial storage allocated for the return variable is deemed insufficient by the QSMRTVBTC API, the API returns the actual size needed in the *Bytes available* subfield of the return variable, and this amount of storage is then reallocated immediately prior to repeating the API call. I've discussed this technique in more detail in the article [APIs by Example: List Open Files API, and the Display Job Open Files Command](#), to which I've included a link at the end of this article.

The third API parameter defines the *Format name* of the receiver variable, and at this point in time only a single format, CBTT0100, is supported. The information returned includes date and time values specifying when the data returned by IBM was retrieved, and the *Time zone* parameter lets you define the time zone to which the date and time values in question should be converted by the QSMRTVBTC API. The fifth and final API parameter is the standard API error data structure, which I assume you're well aware of, and therefore will leave out of scope for now.

Now let's move on to the three new CL commands based on the QSMRTVBTC API. The Work with Cache Battery Status (WRKCCBSTS) command prompt has the following appearance:

```

-----
                                Work with Cache Battery Status (WRKCCBSTS)

Type choices, press Enter.

Operation days . . . . . *NONE          1-999, *NONE
Output . . . . . *          *, *PRINT
-----

```

Specifying a number of days for the *Operation days* parameter, in turn, causes only cache batteries estimated to fail within the specified number of days to be included in the list. The command's help text panel group has all the details. The result of running the WRKCCBSTS command on my system is displayed below:

```

-----
                                Work with IOA Cache Battery Status                                WYNDHAMW
                                                                28-12-11 12:57:39
System name . . . : NOVASTAR          Serial number . . : 65-42CDB
Partition name . . : 65-42CDB

Type options, press Enter.
  5=Display battery data  6=Print battery data

  Resource   Serial      Type      Frame   Card   Battery  Cache
  Opt  Name   Number      Model     ID     Pos.   Maint.  Status
-----
  DC01  YL10D1007006  572F-001  3C02    C1     *YES    *DTAPND
  DC04  YL10D1007006  575C-001  3C02    C2     *YES    *DTAPND
  DC05  YL10P101202F  2BD9-001  3C00    C19    *NO     *DTAPND
  DC06  YL10P10050B7  2BE1-001  3C00    C13    *NO     *DTAPND

                                                                Bottom

Parameters or command
====>
F3=Exit   F4=Prompt   F5=Refresh   F10=Check cache battery status
F11=Battery days  F17=Top      F18=Bottom   F24=More keys
-----

```

Again, comprehensive and cursor-sensitive help text is provided for the list panel and all its parts and columns. Specifying list option 5 will execute the Display IOA Cache Battery Data (DSPCCBDTA) command for the selected cache battery, revealing all the information associated with the cache battery, which the following display panel presents an example of:

```

-----
                                Display IOA Cache Battery Data                                WYNDHAMW
                                                                28-12-11 12:58:08
System name . . . : NOVASTAR          Serial number . . : 65-42CDB
Partition number . : 1                Type/Model . . . : 8202-E4B
Partition name . . : 65-42CDB

I/O Adapter information:

```

```

Resource name . . . . . : DC01
Serial number . . . . . : YL10D1007006
Type/Model . . . . . : 572F-001
Battery type . . . . . : Lithium Ion

```

Operational state:

```

Error state . . . . . : *OPERATING
Maintainable . . . . . : *YES
Cache status . . . . . : *DTAPND
Days to warning . . . . : 778
Days to error . . . . . : 861
Power on time . . . . . : 109
Power on time adj. . . . : 120

```

More...

F3=Exit F5=Refresh F12=Cancel

The final cache battery command I've created to accompany today's APIs by Example article is the Check IOA Cache Battery Status (CHKCCBSTS) command, which offers the following parameters:

Check IOA Cache Battery Status (CHKCCBSTS)

Type choices, press Enter.

```

Operation days . . . . . : 365          1-999
Message queue . . . . . : *USRPRF      Name, *USRPRF, *SYSOPR...
Library . . . . . :                  Name, *LIBL, *CURLIB
+ for more values

```

Just as for the WRKCCBSTS command, the number of days specified for the Operation days parameter will cause warning messages to be sent for all cache batteries estimated to fail within that timeframe. The warning message is sent to up to 10 different message queues specified as the CHKCCBSTS command's second parameter. Here's an example of the warning message being sent:

Additional Message Information

```

Message ID . . . . . : CBX1001      Severity . . . . . : 00
Message type . . . . . : Information
Date sent . . . . . : 28-12-11      Time sent . . . . . : 12:24:46

Message . . . . . : Cache battery failure warning. Battery estimated to fail
in 861 days.
The cache battery associated with the I/O adapter resource named DC01 is
estimated to fail in 861 days. A warning is estimated to be issued in 778
days.

```

To add a job schedule entry ensuring that a check against cache batteries approaching failure within the next 180 days is performed the first day of every month, you would run the following Add Job Schedule Entry (ADDJOBSCDE) command:

```

ADDJOBSCDE JOB (CHKCCBSTS)
          CMD (CHKCCBSTS DAYS(180) MSGQ(*SYSOPR))
          FRQ (*MONTHLY)
          SCDDATE (*MONTHSTR)
          SCDTIME (070000)
          JOBD (*USRPRF)
          JOBQ (*JOBQ)

```

Make sure you adapt the ADDJOBSCDE command's JOBD, JOBQ, and USER parameters to reflect your preferred job runtime settings.

This APIs by Example includes the following sources:

```
CBX242  -- RPGLE  -- Work with IOA Cache Battery Status - CPP
CBX242H -- PNLGRP -- Work with IOA Cache Battery Status - Help
CBX242P -- PNLGRP -- Work with IOA Cache Battery Status - Panel Group
CBX242X -- CMD    -- Work with IOA Cache Battery Status

CBX243  -- RPGLE  -- Display IOA Cache Battery Data - CPP
CBX243H -- PNLGRP -- Display IOA Cache Battery Data - Help
CBX243P -- PNLGRP -- Display IOA Cache Battery Data - Panel Group
CBX243X -- CMD    -- Display IOA Cache Battery Data

CBX244  -- RPGLE  -- Check IOA Cache Battery Status - CPP
CBX244H -- PNLGRP -- Check IOA Cache Battery Status - Help
CBX244X -- CMD    -- Check IOA Cache Battery Status

CBX242M -- CLP    -- Work with IOA Cache Battery Status - Build command
CBX243M -- CLP    -- Display IOA Cache Battery Data - Build command
CBX244M -- CLP    -- Check IOA Cache Battery Status - Build command
```

To create all above command objects, compile and run the CBX242M, CBX243M, and CBX244M CL programs following the instructions in the source headers. You'll also find compilation instructions in the respective source headers of the individual sources.

Note that the QSMRTVBTC API documentation is not yet online at the IBM Information Center. You'll find the QSMRTVBTC API documentation included in the online APARs and PTF cover letters, following the links below.

PTFs Delivering the Retrieve IOA Cache Battery Information (QSMRTVBTC) API:

[7.1 APAR SE45548](#)

[7.1 PTF SI41680 5770SS1](#)

[6.1 APAR SE45547](#)

[6.1 PTF SI41679 5770SS1](#)

[5.4 APAR SE45375](#)

[5.4 PTF SI41535 5770SS1](#)

Related articles and documentation:

[APIs by Example: Zip and Unzip Files with the New 7.1 Zip API Support](#)

[APIs by Example: List Open Files API, and the Display Job Open Files Command](#)

[Carsten's Corner: PTFs and New Command Enable Display of IOA Cache Battery Status](#)

[Retrieve the source code for this API example.](#)

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-ibm-ptfs-new-retrieve-ioa-cache-battery-information-api>