


[print](#) | [close](#)

APIs by Example: Restore from Application API and Exit Program

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 06/26/2008 (All day)

In the preceding installment of APIs by Example ("APIs by Example: The Save to Application API and Exit Program," May 22, 2008), you learned how to save objects to a stream file by using the new Save Object to Stream File (SAVOBJSTMF) command that I presented in that article. If you have installed and tested the SAVOBJSTMF command, you are probably the happy owner of a stream file containing the save records storing the object(s) that you have saved. This is all very fine, though not very useful unless you have an option to reverse the process and restore the saved object(s) from the stream file.

To fulfill that objective, I've written the Restore Object from Stream File (RSTOBJSTMF) command, which incidentally also helps me demonstrate the use of the Restore from Application (QaneRsta) API and the related Restore from Application exit program. Understanding how the involved APIs and exit programs work and interact provides you with the option of adapting or changing the save and restore utilities I present here, to meet your own requirements.

The QaneRsta API has the following parameter list, which is identical to the Save to Application (QaneSava) API's parameter list, except for the fact that all save operation references have been replaced with restore operation references:

1	Qualified user space name	Input	Char(20)
2	User space format name	Input	Char(8)
3	Status format name	Input	Char(8)
4	Status information	Output	Char(*)
5	Length of status information	Input	Binary(4)
6	Error code	I/O	Char(*)

As I explained last time, the first parameter specifies the name and library of a user space that must contain all the control information for the restore operation. This restore control information is formatted as a data structure named SVRS0100, which includes information such as the restore command or API to be used to process the save records, as well as the restore command parameters or restore API parameter key structure. There's also a parameter called *Application data*, and you can use it to communicate between the program calling the QaneRsta API and the exit program called during the restore process.

The name and library of the exit program to be called during the restore process to provide the restore records is also defined by the SVRS0100 data structure. The name of the parameter structure is submitted as the second QaneRsta API parameter. Here's the IBM Info Center's description of all the information available through this parameter:

Offset		Type	Field
Dec	Hex		
0	0	BINARY (4)	Length of structure
4	4	BINARY (4)	Offset to restore command parameters
8	8	BINARY (4)	Length of restore command parameters
12	C	BINARY (4)	Offset to application data
16	10	BINARY (4)	Length of application data
20	14	BINARY (4)	Restore command type
24	18	CHAR (10)	Exit program name
34	22	CHAR (10)	Exit program library
44	2C	CHAR (8)	Target release
		CHAR (*)	Restore command parameters
		CHAR (*)	Application data

The restore commands and APIs that the QaneRsta API supports are defined by the following list, which shows all available options at release V5R4 for the Restore command type subfield in the preceding structure:

1	Restore (RST) command
2	Restore Object (RSTOBJ) command
3	Restore Document Library Object (RSTDLO) command
4	Restore Library (RSTLIB) command
5	Restore Object (QsrRestore) API
6	Restore Object List (QSRRESTO) API
7	Restore System Information (RSTSYSINF) command

The QanaRsta API's third, fourth, and fifth parameters define the name of the status data structure, the status data structure itself, and the status data structure length, respectively. The status data structure is named SRSTO100, is identical to the equivalent QaneSava parameter, and provides a communication area between the QaneRsta API and its caller. For release V5R4, the following information is accessible through the status data structure:

Offset		Type	Field
Dec	Hex		
0	0	BINARY (4)	Bytes returned
4	4	BINARY (4)	Bytes available
8	8	BINARY (4)	Transfer time
12	C	BINARY (4)	Transfer block size
16	10	BINARY (4)	Transfer block multiplier
20	14	BINARY (4)	Last block size
24	18	CHAR (10)	User space library used
34	22	CHAR (2)	Reserved
36	24	BINARY (4)	Decimal transfer time

The final sixth parameter is the standard API error code data structure. Similar to the QaneSava API, the QaneRsta API has many restrictions, constraints, and subtleties to consider when dealing with this API. For example, you can restore objects only if these were previously saved by the QaneSava API and only if the objects were saved from the current or an earlier release of the operating system.

Additionally, because the restore command is processed from within a prestart job, you cannot rely on adopted authority to execute the specified restore command. One way to provide sufficient authority to the prestart job is to use adopted authority to swap user profiles before calling the

QaneRsta API. At the end of this article, I include a link to the IBM V5R4 version of the QaneRsta API documentation. Be sure to check out all the details.

Until now, practically everything I've explained about the restore process has been a mirror image of my article that explains the QaneSava API and exit program. And, as you will see, the similarities continue when it comes to the documentation of the Restore from Application exit program.

Following a successful call to the QaneRsta API, the Restore from Application exit program is called during any of the following four operation events:

1. Start of restore process. The exit program must prepare for the restore record transfer -- for example, open the stream file, in which the records are stored.
2. Transfer of a block of restore records. The exit program returns a block of restore records as well as the length of the data block. This step must be repeated until all restore records have been returned.
3. End of restore process. The exit program at this point must terminate the process of returning the restore records, and, for example, close the stream file.
4. Abnormal end of save process. If anything goes wrong in the course of the restore process, the exit program is called one final time to allow it to perform cleanup and termination activities -- for example, close the restore records stream file.

The exit program has the following required parameter group:

1	Operation type	Input	Binary(4)
2	Operation status	Output	Binary(4)
3	Restore data	Input	PTR(SPP)
4	Length of restore data	Input	Binary(4)
5	Restore bytes written	Output	Binary(4)
6	Qualified user space name	Input	Char(20)
7	User space format name	Input	Char(8)

The *Operation type* parameter specifies which of the preceding four events caused the exit program to be called, and the exit program returns the outcome of the processing performed in the Operation status parameter, subsequently causing the restore process either to continue or terminate.

The third parameter is a space pointer to a block of restore records. This parameter is of course only used by the restore process when the operation type is 2. The *Length of restore data* parameter ensures that the restore process addresses only the restore data provided by the restore exit program. The sixth parameter is the qualified name of the user space specified as input to the QaneRsta API, and it thereby enables the exit program to share and exchange information with the API caller via the specified user space. The seventh parameter declares the format of the data in the user space.

Putting all this information to work provides you with the knowledge and components necessary to build the RSTOBJSTMF command, and once the command is completed, here's what you'll see when you prompt the RSTOBJSTMF command:

```

Restore Obj from Stream File (RSTOBJSTMF)

Type choices, press Enter.
```

```

Objects . . . . . Name, generic*,
*ALL
      + for more values

Library . . . . . Name

Object types . . . . . *ALL
      + for more values

Stream file . . . . .

Additional Parameters

Option . . . . . *ALL      *ALL, *NEW, *OLD,
*FREE
Data base member option . . . . *MATCH      *MATCH, *ALL,
*NEW, *OLD
Allow object differences . . . . *NONE      *NONE, *ALL,
*AUTL...
      + for more values

Restore to library . . . . . *SAVLIB      Name, *SAVLIB
Output . . . . . *NONE      *NONE, *PRINT

```

The *Stream file* parameter specifies the path to an existing stream file containing the restore records previously obtained by the SAVOBJSTMF command or a similar facility based on the Save to Application API and exit program. As always, a help text panel group is included with the command to explain all details and parameters. Apart from the Stream file parameter, all other parameters work the same as they do in the Restore Object (RSTOBJ) command.

Please note that if you specify the OUTPUT(*PRINT) keyword for the RSTOBJSTMF command, you can locate the produced spooled file by using the command:

```
WRKSPLF SELECT(*CURRENT *ALL *ALL *ALL *ALL QPSRLDSP)
```

Because the restore process runs in a separate prestart job, the spooled file is created under the QPRTJOB special job name and is consequently not owned by the job running the RSTOBJSTMF command, thus requiring extra effort to locate the printed output.

And because both the save and restore process is performed through a prestart job when you run the SAVOBJSTMF and RSTOBJSTMF commands, you could find yourself in a situation in which you'd want to locate that job. One way to achieve that objective is to use the Display Object Description (DSPOBJD) command against the saved and restored objects to acquire the information necessary to look up the prestart job in question. For a saved or restored object, run the following command:

```
DSPOBJD OBJ(QGPL/ABC) OBJTYPE(*FILE) DETAIL(*FULL)
```

Page down until you get to the last page with the object save/restore information:

Display Object Description - Full			
Library 1 of 1			
Object	: ABC	Attribute	:
PF			
Library	: QGPL	Owner	:
CARSTEN			
Library ASP device . .	: *SYSBAS	Library ASP group . .	:
*SYSBAS			
Type	: *FILE	Primary group	:
*NONE			
Save/Restore information:			
Save date/time	:	22-06-08 10:27:02	
Restore date/time	:	17-05-08 13:22:16	
Save command	:	SAVOBJ	
Device type	:	Save file	
Save file	:	QTEMP/QANE738456	

Notice the six digits appended to the save file name QTEMP/QANEnnnnnn. This is the job number for the prestart job named QANEAGNT that performed the save or restore operation against the specified object. Using that job number as well as user profile QUSER with the Work with Job (WRKJOB) command, brings up the Work with Job panel for the save or restore job:

```
WRKJOB JOB(738456/QUSER/QANEAGNT)
```

From this panel, you can, for example, look up the job log to examine the messages issued as part of the save or restore process to verify the successful execution or investigate any problems encountered.

Previously published related articles:

APIs by Example: The Save to Application API and Exit Program (May 22, 2008)

<http://systeminetwork.com/article/apis-example-save-application-api-and-exit-program>

This article demonstrates the following Backup and Recovery API and Exit Program:

Restore from Application (QaneRsta) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/apis/QaneRsta.htm>

Restore from Application Exit Program:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/apis/XANERSTA.htm>

Backup and Recovery APIs V5R4:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/apis/back1.htm>

You can retrieve the source code for this API example from:

http://www.pentontech.com/IBMContent/Documents/article/56846_628_RstObjStmf.zip.

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-restore-application-api-and-exit-program>