

## Freeing your IBM i libraries and files

### Eliminate tape media incompatibilities with save files and virtual tape

Eduardo Delgado

January 09, 2012

Long time users of IBM® Power Systems® hardware may find themselves stuck with tape drives that are nearing obsolescence. Quarter-inch cartridges, 8mm, and reel-to-reel tape drives don't have much of a place in the IBM POWER6® and POWER7® world. Learn how you can use disk space to back up IBM i objects in a format that can be transmitted to any other IBM i system without using a common tape media, saving objects to save files or virtual tapes, and then using FTP to transmit them to another IBM i system or Windows disk drive.

### The value of tape media independence

This article addresses a problem when trying to move data from an older IBM® AS/400® family server to a newer IBM Power® System running IBM i—namely, how do you restore a data file from an old tape cartridge to a system that doesn't have a compatible tape drive? The techniques described here not only make it easy to circumvent incompatible tape drives but may also save thousands of dollars when planning to replace an older midrange server with a new Power System.

### Tape compatibility problems

This dilemma results from the plethora of different tape media that IBM has supported since the announcement of AS/400 technology in 1988. AS/400 hardware is quite durable, so many of these old tape drives are still in active duty. Faster tape drives have since been introduced that hold more data but whose cartridges are physically different from their predecessors.

Over the years the Power Systems family of computers has supported tape drives such as:

- Reel-to-Reel tape (in 1988, only companies with large budgets could afford these for their backups)
- QIC tapes of multiple densities
- 3480s and 3490s
- 3570s (with their unique trapezoid shape)
- 8mm tapes of various densities

- LTO-1 through LTO-5
- Digital Audio Tape (DAT) DAT72 and DAT160

Many midrange systems installed by small and medium businesses would have an internal QIC tape drive, but most of the other technologies required a cable connection through an I/O adapter (IOA). Newer Power Systems support mainly LTO or DAT tape drives. The architecture of the newer hardware is geared to faster peripherals. Therefore, the adapters needed for old technologies like 8mm and IBM 3570s simply won't work on the IBM POWER7 system.

## Considerations for upgrades and migrations to new Power Servers

Companies looking to upgrade from their old IBM OS/400® servers to a new POWER7 server must figure out how to move hundreds of gigabytes of data without a common tape drive. It may be possible to add an LTO tape drive to the old server, but spending money to add resources to hardware they intend to displace seems wasteful. There must be a viable way to transport data—be it a single file or all the user data—from one system to another without requiring tape. Actually, there are two: save files and virtual tape.

## Save file technology

Save files have been part of the IBM i operating system since 1988. These special file types, stored in libraries, emulate tape drives. Instead of saving a library or object to a tape device, you would specify the name of the save file as the target. Therefore, the information gets written to disk instead of to tape. When the Save operation is complete, the contents of the save file can be displayed or restored just like the contents of a loaded cartridge. The key feature is that save file objects can be shipped to another IBM i server or to a Windows® device using FTP.

## Benefits of save files

Initially, save files were used to provide a shorter window for backing up critical data: Backing up a library to disk drives was sure to be faster than writing to a comparatively slower tape drive. Then, when the backup was complete, the save files could be saved to tape at the administrator's leisure for disaster recovery purposes. This remains true, but the ability to transfer entire save files between systems makes them a powerful tool for distributing objects without using tape at all. Once a save file has been sent over FTP to a Windows file (with the extension *.savf*), it can be attached to email messages, copied to USB drives, and archived to huge file servers. In fact, software vendors commonly deliver their code in a .zip file containing a save file instead of mailing physical media.

## Creating a save file

The steps for using save files are relatively simple. And, because they use Command Language (CL) commands, you can easily collect the rules into a program for automation.

### Step 1. Create the save file in a library

You save the save file in a library using the `CRTSAVF` command. `CRTSAVF FILE(QGPL/SAVF1)` creates the save file SAVF1 in the QGPL library.

## Step 2. Save the desired objects to the save file

An important limitation comes into play here. A single save file can only be used to save objects within one library or, if working in the Integrated File System (IFS), the contents of one directory (including its subdirectories). For this example, a programmer's source code files are in the RPLIB library. You can save the entire library to the save file using the `SAVLIB LIB(RPLIB) DEV(*SAVF) SAVF(QGPL/SAVF1)` command.

Instead of listing a tape drive address for the `DEV` parameter, you use the value `*SAVF`. The command will need to include a valid save file name and library for the `SAVF` parameter, or an error message appears.

### Saving objects in one library

Individual objects within one library could have been saved using the `SAVOBJ` command instead of `SAVLIB`. All the normal save rules apply here—namely, the objects being saved must not be locked by another job, and users must have permission for the objects they are saving. If the intent is to restore the contents of the save file to a system at a lower release level, you would need to specify that level in the target release parameter (`TGTRLS`).

A good rule of thumb is to expect that the save file will require as much disk space as the library or objects that it will contain. These techniques are undesirable if the systems are short on available space. However, compression options are available. For example, you can set the data compression parameter (`DTACPR`) to `*LOW`, `*MEDIUM`, or `*HIGH`. The default value is `*LOW`, but `*MEDIUM` and `*HIGH` can save more space at the expense of performance.

Finally, you can use save files to back up objects within an IFS directory. The difference, however, is that you must use the `SAV` command. This command was designed to back up objects in the IFS and thus has a "UNIX®" look to it. The notation demands a fully qualified location for the backup device. Therefore, to save a directory called `/samples` with all its subdirectories to the `SAVF1` save file, you would use the command `SAV DEV('QSYS.LIB/QGPL.LIB/SAVF1.FILE') OBJ('/samples' *INCLUDE))`

You can verify that the objects were successfully copied by displaying the save file just as you would display the contents of a tape cartridge. Instead of using the `DSPTAP` command however, you would use `DSPSAVF FILE(QGPL/SAVF1)`. the following:

## Step 3. Use FTP to send the save file to a Windows PC file or another IBM i system

FTP is effective for downloading save files to PC files or to other Power Systems. The target for the transmission will determine the specific FTP commands you use. When the file is sent to another IBM i machine, it will appear as an object in a library and can be displayed or restored using normal CL commands. However, when the save file is sent to a Windows disk, it exists as a `.savf` object, which is a nothing more than a blob of bytes to Windows. The file can only be useful if it finds its way to a Power server.

Perform the following steps to copy the save file created in [step 2](#) to a PC USB drive (labeled `F:\` in this example):

1. Enter the following information from a DOS prompt on the target PC:
  - **FTP nn.nn.nn.nn:** The IP address or a valid network name for the source system
  - **User profile:** An authorized profile on the source system
  - **Password:** The corresponding password
2. Enter these FTP commands to create the SAVF1 save file in drive F on the PC:
  - BIN
  - QUOTE SITE NAMEFMT 1
  - GET /QSYS.LIB/QGPL.LIB/SAVF1.SAVF F:\SAVF1.SAVF
3. When the transfer is complete and successful, end the session with the QUIT command.

If the source and target IBM i systems can communicate over a network, you can transmit the save file directly from one to the other. To get the save file created in [step 2](#) directly into a library on another Power server, perform these steps from the target system:

1. Enter the following information from a command line on the target Power server:
  - **FTP nn.nn.nn.nn:** The IP address or a valid network name for the source system
  - **User profile:** An authorized profile on the source system
  - **Password:** The corresponding password
2. Enter these FTP commands to create or replace the SAVF1 save file in the QGPL library on the target system:
  - BIN
  - NAMEFMT 1
  - GET /QSYS.LIB/QGPL.LIB/SAVF1.SAVF /QSYS.LIB/QGPL.LIB/SAVF1.SAVF
3. When the transfer is complete and successful, end the session with the QUIT command.

Use the DSPSAVF command to prove that the contents of the save file are valid.

## Restoring objects in a save file to another Power server

You can transfer a save file residing on a PC hard drive or a USB drive to a Power server by using FTP over a network. Because the save file in the earlier steps wound up on drive F with the name SAVF1.SAVF, perform these steps to move it to the target server:

1. Enter the following information from a DOS prompt on the target PC:
  - **FTP nn.nn.nn.nn:** The IP address or a valid network name for the target IBM i system
  - **User profile:** An authorized profile on the source system
  - **Password:** The corresponding password
2. Enter these FTP commands to create or replace the SAVF1 save file in the QGPL library of the target system:
  - BIN
  - QUOTE SITE NAMEFMT 1
  - PUT F:\SAVF1.SAVF /QSYS.LIB/QGPL.LIB/SAVF1.SAVF
3. When the transfer is complete and successful, end the session with the QUIT command.

From there, you can restore the entire library using the RSTLIB SAVLIB(RPGLIB) DEV(\*SAVF) SAVF(QGPL/SAVF1) command.:

The `RSTOBJ` command restores individual files or objects within the library. This process parallels the approach used by physical tape cartridges, but the need for tape media has been totally circumvented.

## Limitations of save files

Save files are a relatively quick and simple way to move libraries around, but their use isn't without caveats. The main limitation is that only one library—or objects within one library—can be saved at a time. The size of a single save file cannot exceed 2TB, and saving hundreds of gigabytes of data to a save file might create a performance bottleneck on the disk drives. Using an independent application service provider to direct the traffic to specific disks would be advisable in that situation. Performance testing is always recommended.

## Virtual tape solutions

### Before setting up virtual tape

IBM has published a 500-page Redbook to explain this (see Resources for a link). There's no need to read the entire manual to get started, but the book gives good advice on planning for the space requirements of the image catalogs.

Also, although it's possible to save the entire system to a virtual tape, the image from such a backup cannot be used to reload the operating system in case of a disaster. Carefully review the Redbook if virtual tapes become a key element of the regular backup procedures.

Version 5 release 4 of IBM i5/OS® introduced virtual tape technology to allow a disk area to behave like a tape drive. Therefore, the full compliment of save commands is available to back up anything from a single object to the complete contents of the server.

Although it's possible to stack a bunch of save file commands in a program to back up multiple libraries, it's more efficient to use one save command and one target (the virtual tape drive). The data will be written to a virtual tape image catalog that resides in the IFS, not in a library like the save files. Thus, an image catalog containing a group of disparate objects can be moved to other locations just like any other IFS directory.

Now, you can transmit a huge amount of data from one Power System to another via FTP instead of using tape cartridges. This technology provides a useful option when migrating data from older systems (running at least version 5 release 4) to newer systems. Although having a common tape drive is always easier, virtual tapes may save the cost of having to acquire a tape drive and IOA for the old server.

Implementing virtual tape is not without its hazards. Because a lot of data is being saved to the IFS, ample disk space must be available. A system whose disks are 80% full is not a good candidate. In addition, setting up and managing the image catalogs can be complicated. For example, when the image catalog is created, it is possible to allocate disk space to it immediately, which can reserve hundreds of gigabytes of space, even if it may not be needed.

## Creating and restoring a virtual tape and image catalog

The exact commands for implementing virtual tapes are beyond the scope of this article. [Table 1](#) provides a high-level procedure to create and download the catalog.

**Table 1. Summary of creating a virtual tape and image catalog**

Step	Process	Key CL commands
1.	Create the virtual tape drive.	CRTDEVTAP, VRYCFG
2.	Create an image catalog and image catalog entry.	CRTIMGCLG, ADDIMGCLGE, LODIMGCLG
3.	Save the desired data to the virtual tape.	SAVSYS, SAVLIB, SAVOBJ
4.	Use FTP to transfer the contents of the image catalog entry directory to a PC file.	FTP

To complete the scenario, the image catalog entry objects will be transferred from a Windows directory to an IFS directory on the target server. From there, you will need to create a new image catalog and import the objects to create the entry. [Table 2](#) provides an overview of the steps involved.

**Table 2. Restoring an image catalog entry and creating a new image catalog from it**

Step	Process	Key CL commands
1.	Use FTP to transfer the image catalog entry directory objects to an IFS directory.	FTP
2.	Create a new image catalog using the uploaded directory.	CRTIMGCLG, LODIMGCLG

From there, the objects can be restored using normal restore commands.

## Conclusion

Although tape drives remain a fundamental part of disaster recovery plans, they are becoming less critical for transferring data to other systems. As disk technology and storage area networks become more elegant, using save files and virtual tape drives will become a more appealing option for moving data off of the host server.

## Resources

### Learn

- For more information on planning a virtual tape solution, read the IBM Redbook [i5/OS V5R4 Virtual Tape: A Guide to Planning and Implementation](#).
- Learn more about save files for [IBM i 6.1](#) and [IBM i 7.1](#) from the IBM i Information Center.
- Learn more about virtual tape media for [IBM i 6.1](#) and [IBM i 7.1](#) from the IBM i Information Center.

## Discuss

- Follow [developerWorks on Twitter](#).
- [developerWorks IBM i blogs](#): Check out our blogs and get involved in the [developerWorks IBM i community](#).

© Copyright IBM Corporation 2012

([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

[Trademarks](#)

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))