

# Binder language export source in stream files

## Processing export source directly out of stream files

Mark Hessler

May 09, 2016

Starting in IBM i 7.3, the `CRTSRVPGM`, `UPDSRVPGM`, and `RTVBNDSRC` commands now process stream files containing binder language export source. With this enhancement, the binder language source can now reside in either a stream file or a physical file. This article describes this new support and how it can be used.

### Introduction

Processing resources directly out of stream files is something that IBM i customers have been requesting for a number of years. The enhancement in version 7.3 puts IBM i closer to that goal. When creating or updating a service program, the export source specifies the service program's signature and the names of procedures and data that the service program exports. Specifying the file is done using the `EXPORT` parameter on the `CRTSRVPGM` (Create Service Program) and `UPDSRVPGM` (Update Service Program) commands. In the past, this source could only be processed from a physical file. Starting in IBM i 7.3, the binder language export source can be processed from a stream file.

### `CRTSRVPGM` command

Creating a service program using the `CRTSRVPGM` command requires specifying the file name containing the export source. In releases before IBM i 7.3, the command offered the `SRCFILE` and `SRCMBR` options enabling users to provide the name of a physical file and member containing the source. Supporting the `SRCSTMF` option, the command now can be used to identify any path to a stream file (for example, `/home/company/product/file`) in the integrated file system.

### `UPDSRVPGM` command

After creating a service program, it can be updated using the `UPDSRVPGM` command. Just like the `CRTSRVPGM` command, the `SRCSTMF` option can be used on the `UPDSRVPGM` command to read the export source from a stream file. In fact, even if the export source was originally read from a physical file, the `UPDSRVPGM` command can be used to read the export source from a stream file. For example, if a service program is created with the export source from a physical file, `SRCLIB/QSRVSRC(SPGM)`, using the following command:

```
CRTSRVPGM SRVPGM(SPGMLIB/SPGM)SRCFILE(SRCLIB/QSRVSRC) SRCMBR(SPGM) ...
```

After the export source is copied to the stream file `/binder/source/spgm.src`, the service program can be updated to obtain the export source from that location by issuing the following command:

```
UPDSRVPGM SRVPGM(SPGMLIB/SPGM) EXPORT(*SRCFILE)
SRCSTMF('/binder/source/spgm.src') ...
```

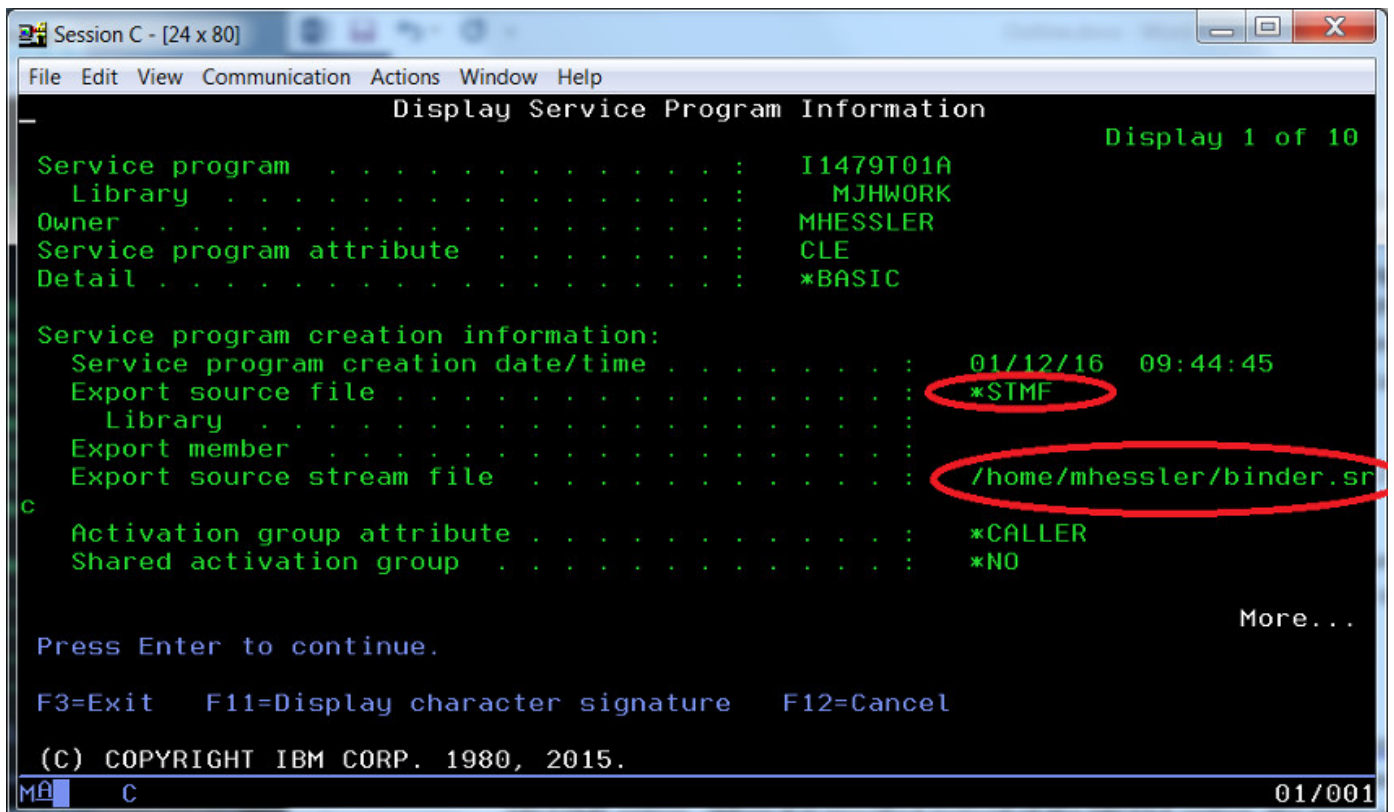
### RTVBNDSRC command

When setting up the initial binder language for a service program, the `RTVBNDSRC` (Retrieve Binder Source) command can be used. The command can be run against the list of all modules that will make up the service program to create a binder source file containing every export in the list of modules. The binder language source can then be modified to remove the names of the procedures and data that should not be exported by the service program when it is created. The `RTVBNDSRC` command has also been updated to support the `SRCSTMF` option. By specifying this option instead of the `SRCFILE` option, the generated export source is placed in the given stream file. You can then use this stream file in the `SRCSTMF` option of the `CRTSRVPGM` command when creating the service program.

### DSPSRVPGM command

The `DSPSRVPGM` (Display Service Program) command indicates the export source file used when the service program was created or updated. The command has been modified to show which type of export source file was chosen, either a physical file or a stream file. Figure 1 shows a sample output of the `DSPSRVPGM` command when the export source is read from a stream file.

Figure 1. DSPSRVPGM panel



If the binder source came from a physical file, the *Export source file* field displays the file name as it did before. If the binder source came from a stream file, the *Export source file* field will display the string, \*STMF. This indicates that the path to the binder source file is listed three lines below in the *Export source stream file* field. This field will always show the absolute path to the stream file, even if a relative path name is given in the `CRTSRVPGM` or `UPDSRVPGM` command. This way the location from where the export source was obtained can be precisely determined.

## QBNRSPGM application programming interface (API)

The `QBNRSPGM` API is a way to get the same information programmatically that would be obtained from the `DSPSRVPGM` command. The `SPGI0100` format is updated to return a structure that includes the path of the stream file, if one was used for the export source when the service program was created. If a stream file was not used, the structure returned is unchanged and therefore the existing programs that use this API do not have to change.

If a stream file was specified when the service program was created or updated, the structure returned from `QBNRSPGM` will be changed as follows:

- The stream file's absolute path name and path name length will be included at offset 444 and 440 of the structure, respectively.
- To indicate that the structure returned includes an absolute path name and length, the *export source file name* field at offset 61 will contain the special value, \*STMF. The *export source library name* and *export source member name* fields will contain blank values in that case.
- The absolute path name at offset 444 will be encoded in the CCSID of the job at the time the service program was created. That CCSID will be stored in the structure at offset 436.

Note that even if a program using this API was not changed, it might still work if used against a service program created with the export source obtained from a stream file. The only difference the program calling the API would see is that the *export source file name* field contains the string, \*STMF.

## Benefits

With this enhancement, the export source file can be stored in the same location as the rest of the application source code and used directly from that location. In the past, the export source would need to be stored in a physical file or copied to a physical file before it could be processed. When the stream file name is specified on the `CRTSRVPGM`, `UPDSRVPGM`, or `RTVBNDSRC` commands, it can be specified either as an absolute name or a relative name. An absolute name starts with the directory separator character (usually the '/' character) and relative names do not start with this character. The path shown with the `DSPSRVPGM` command will always show the absolute path to the stream file. That way there is no confusion from where the file was obtained. As always, files from the `QSYS.LIB` file system can be specified as a path in the `SRCSTMF` option. The option `SRCSTMF ('/QSYS.LIB/XXX.LIB/YYY.FILE/ZZZ.MBR')` will get the export source file from the physical file `XXX/YYY.ZZZ`. However, because the export source is specified as a stream file and not a physical file, the `DSPSRVPGM` command will display the file name as a path in the *Export source stream file* field, and not as a physical file in the *Export source file* and *Export source member* fields.

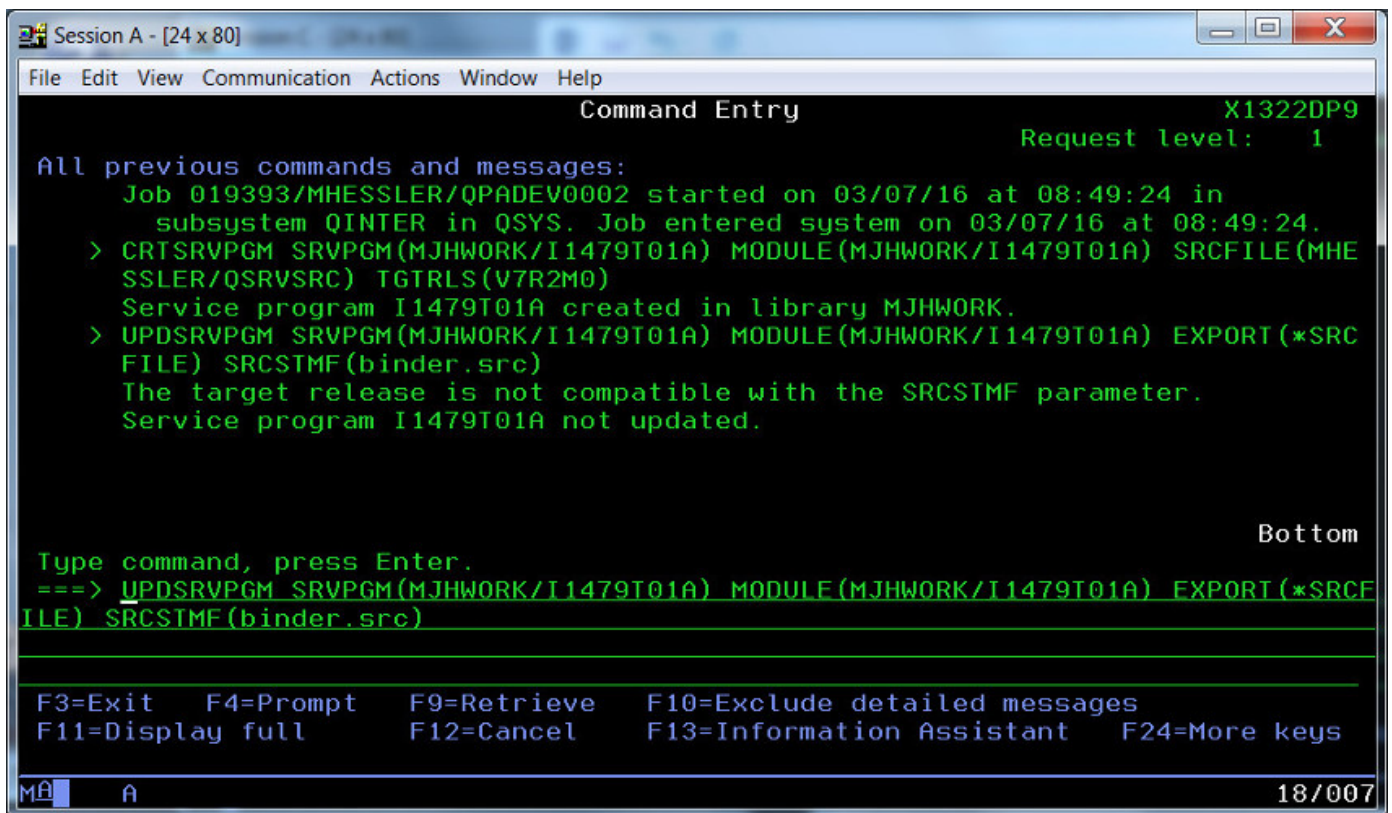
## Cautionary statements

A few cautionary statements should be made regarding this new support. First, because relative names can be specified, the actual locations of the export source file might be confusing. If the relative stream file *binder.source* is specified and the file is expected to be located in */company-name/product/binder.source*, then make sure that the current working directory is set to */company-name/product* using the `CHGCURDIR` (Change Current Directory) command.

Second, if the file specified is */company-name/product/binder.source*, ensure that the profile has execute (\*X) authority to each of the directories in the specified path (*company-name* and *product*). In addition, read (\*R) authority is needed to the stream file for the `CRTSRVPGM` and `UPDSRVPGM` commands and write (\*W) authority is needed to the stream file for the `RTVBNDSRC` command.

Lastly, this support is only allowed for objects that have a target release of V7R3M0 or later. If the `CRTSRVPGM` command is used with `TGTRLS(V7R2M0)` or `TGTRLS(V7R1M0)`, the `SRCSTMF` option cannot be specified. If the `UPDSRVPGM` command is used on an object with a target release before V7R3M0, the `SRCSTMF` option cannot be used. Diagnostic message CPD9390 will be issued. For example, Figure 2 shows the service program I1479T01A being created with a target release of V7R2M0. When the service program is updated to obtain the export source from a stream file, error CPD9390 is generated, as shown in Figure 2.

**Figure 2. CPD9390 message**



## Summary

Being able to contain all programming source in the integrated file system can help to streamline the process of creating and maintaining service programs on IBM i. The export source can now be handled directly from stream files. It no longer needs to be stored in or copied to a physical file before it can be used.

## Resources

- Refer to IBM i 7.3 in the [Knowledge Center](#).
- Refer to information on the `CRTSRVPGM`, `UPDSRVPGM`, `RTVBNDSRC`, and `DSPSRVPGM` commands in the [IBM i 7.3 Knowledge Center](#).
- Refer to information on the [QBNRSPGM API](#) in the [IBM i 7.3 Knowledge Center](#).
- The [ILE Concepts](#) manual in [IBM i 7.3 Knowledge Center](#) provides information about the Integrated Language Environment service program creation process.

© Copyright IBM Corporation 2016

([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/))