6.1 — Git 'er Done

By Garth Tucker

I've been involved in many, many releases of the OS for our beloved platform over the years and for the most part have always been happy with the new features, improvements in performance, and/or the updating of existing LPPs, but (in my more than humble opinion) 6.1 has topped them all!

One of my favorite enhancements is the option of DVD media—almost totally removing the need to sit there flipping CDs while installing PTFs or creating image catalogues. This may come across as being a little frivolous and it's not that I'm lazy (or at least not that lazy), but there's room for error when swapping CDs, and who needs that extra risk during an upgrade or cutover?

What really sets this version apart though, is the overall improvement in performance, integrity, and functionality. This is associated with the new hardware instruction sequences and new internal structure. Converted programs still perform the same machine interface operations, however the internal format of the program object is changed and the executable hardware instructions are replaced. Conversion might be better described as applying changes to objects and "upgrade" or "refresh" might be more relevant terms. Machine Interface programs will retain their name, location, owning user profile, and other such attributes.

Now that we've established what is causing most of the buzz surrounding 6.1, what exactly is program conversion? Is it like CISC to RISC? Sort of... Is it difficult? Sort of... Is this upgrade something I should do? Most certainly!

Program conversion starts at your current release, V5R3 or V5R4 with installation of the PTFs to allow for the ANZOBJCVN (analyze object conversion) command. You must make yourself aware of the following terms that are used in the reports generated by the ANZOBJCVN command;

- State
- Digital signatures
- Profiling

From the much bigger brains than mine at IBM Rochester, here's what those terms mean to us.

State:

MI programs run with either user or system state. This state attribute controls whether the program has any access to system domain objects at high security levels. State also determines the type of access (no access, read only, or reads and writes) that a program has to user domain spaces. Application programs run with user state, most operating system programs run with system state.

Digital signatures:

Digital signatures are used to ensure that an object has not been modified and that it came from the expected source. Most often, digital signatures are used as an object integrity check before the object is restored onto a system. Digital signatures can optionally be applied to several types of objects, including programs. They must be removed during program conversion because the program is changed, thus invalidating any previous digital signature.

Profiling:

Application profiling is an advanced optimization that utilizes data-specific information to further improve program performance. A profiled program is reorganized according to how frequently its control-flow paths are run when operating on data that is considered typical of the data it will process when in production.

Now that we are aware of what program conversion is, we can look at the high-level steps required to perform the function.

First, collect information about objects on your system. Run the ANZOBJCVN OPTION (*COLLECT) command, each collection of a given type replaces the data in the collection file(s) for that type.

Next, review and understand collected information. Run the ANZOBJCVN OPTION (*REPORT) command, and optionally run your own queries over the collected data. Consider the following reports:

- Report on programs that won't convert with the *CVNPRB parameter.
- Report summary information, such as estimated conversion times for each library, with the *LIBSUM parameter.

Then, contact your application provider, if necessary. If you find you have a purchased application that won't convert and you don't have the source code to recompile it, contact the application provider to get a version of the application that can run on V6R1.

Finally, remove unsupported products, then upgrade to 6.1. Remove unsupported Licensed Program Products (LPPs) and other unsupported products before migrating to V6R1. Once all the preparation tasks for object conversion are completed check the InfoCenter topic "Installing, upgrading, or deleting i5/OS and related software."

Converting Objects in 6.1

There are three methods for converting your systems objects and you may choose the combination of them that best suits your requirements.

1. "Conversion using STROBJCVN":
If we choose to convert using STROBJCVN, we can submit the job to batch and let it run in the background, it will go through the system and touch each object and thereby convert it for use at 6.1. This would be my preferred way of converting in most situations.

2. "Conversion of program objects on restore":
The second method would be applicable during a RISC to RISC migration, but think about how long your restore is going to take if every object is converted during it's restore onto the system.

3. "Conversion the first time a program is run or called":
For those of us who have extremely tight windows for the upgrade, the third method combined with the first would be our best bet for getting back into production as quickly as possible. It would make response times a bit slower for users initially as every time they attempt to access an object that has not been converted, it would perform the function. Is it going to make the system seem like it's a pig? No, the additional response times will not be that bad. It would be important to submit the STROBJCVN command to batch as well to help the process move along and finalize the conversions more quickly.

Continued on page 20 — "Git 'er Done"

Don't let this be you!

Cost Effective Continuity Solutions

♦AS/400, iSeries & i5 ♦Wintel & Linux
♦Unix ♦AIX

Business Continuity Solutions, as well as Disaster Recovery Hot-Site

Dynamic DR

905-841-9891
sales@dynamicdr.com
www.dynamicdr.com

—"Git 'er Done" Continued from page 15

Integrity and What it Means to Us

As mentioned earlier, integrity of the system has been enhanced and this is due to 6.1 conversions removing any altered programs that exist now, extending the ability to remove any future code corruption on any runnable MI application without program source and requires no ongoing updates. It prevents the loading of non-i5/OS "system state" programs and stops the loading of non-i5/OS programs that lack creation data. Conversion removes any unsupported alterations by creating the program using only defined MI constructs.

Unlike anti-virus software on other platforms, MI updates are not required to stay ahead of new forms of attack. Instead of constantly updating anti-virus software, re-creating a program from its MI constructs, as specified in its creation data, will always eliminate threats. Once again, how did these other platforms get so popular? Here we are with an OS that by design/nature does not allow destructive software to run, but yet people still insist on running software that is open to malicious intent from virii and hackers. Guess it's true what they say; you cannot save people from themselves.

Taken as a whole, this upgrade may seem a little intimidating, but proper planning and ensuring that you have all your ducks in a row will allow you to move to 6.1 and begin enjoying the many other new features, functions and enhancements that we didn't have time to cover in this short article. For more information on those topics, contact your IBM rep or IBM Business Partner rep or have a look through the IBM website for details.

***Garth Tucker**, managing director of i3 Tech Group and Dynamic Disaster Recovery of Aurora, is an IBM System i Specialist and Business Continuity Planner, IBM Certified for V4R3 through V5R4 as well as being CompTIA Linux+ Certified. He has many years of experience with System i and helped write the Technical Overviews of OS400; V4R4, V4R5, V5R1 and V6R1 with the IBM ITSO. In addition, he has presented sessions at COMMON, TEC conferences and TUG MoMs.*