

Nov 22 '10

The DB2 Catalog Gets a Makeover

by Willie Favero in [z/Journal](#)

When DB2 10 for z/OS arrives, it will usher in a host of enhancements to both the DB2 catalog and directory. This article addresses how to prepare for migration and details of all the new or improved enhancements. Note that since IBM hasn't yet announced the Generally Available (GA) date for DB2 10, some things covered here could change, and the timing of this article means it can't cover every possible new feature or change.

Preparing for Migration

Before getting into the DB2 10 enhancement details, let's first discuss how to prepare to make all these changes. During your catalog migration to DB2 10 Conversion Mode (CM), there are two "optional" activities you should perform. Both steps validate the migration worthiness of the catalog and really aren't optional.

The first optional job step, Migration Step 2, runs the standalone DB2 utility DSN1CHKR, checking the five catalog table spaces that use links and one directory table space that uses hashes for broken links, broken hash chains, and orphans. These catalog and directory table spaces will have their links and hashes removed while being converted to universal table spaces during the Enabling New Function Mode (ENFM) process. The ENFM conversion process can only be more successful knowing that the inputs to the conversion are clear of problems.

Also during this job step, DSN1COPY with the CHECK option should be run against all catalog and directory table spaces. This will verify that all the catalog and directory table space pages are physically correct and that the catalog table spaces are clustered. DB2 should be stopped (or, at a minimum, have all activity disabled), while you run DSN1CHKR and DSN1COPY to avoid errors caused by some catalog or directory pages still being in the buffer pools. Both of these standalone utilities run against the disk copy of the data only and aren't aware of any pages in the buffer pools.

Finally, the third optional job in Migration Step 2 you should run is the CHECK INDEX utility for all DB2 catalog and directory indexes. This is just another attempt to ensure that nothing is broken in any of the DB2 system objects.

The second optional step that shouldn't be skipped or ignored is Migration Step 4. A set of queries in member DSNTESQ in SDSNSAMP can be used to verify the logical correctness of the catalog tables. Although you can run these queries with SQL Processing Using File Input (SPUFI) or DSNTDP2, you should use DSNTDP2 so you can retain a record of the outputs.

The migration process requires an image copy of the catalog and directory. Having a backup of all the catalog and directory table spaces is critical to ensure recoverability to a previous version of DB2. Don't take chances with your catalog and directory by skipping this step.

DFSMS Mandatory

Data Facility Storage Management System (DFSMS) made its debut back in 1993 and there are still DB2 professionals unwilling to use it with DB2. They've even had incentives; every release of DB2 adds a few more features dependent on DFSMS. But now, DB2 10 may have come up with the biggest incentive of all. DFSMS, or DB2-managed storage, is mandatory before a DB2 10 upgrade or new installation can even be considered, before even getting to CM.

—
—

DFSMS is a set of IBM software products that automatically manages your data from the point where you create it until you no longer need it (or move it to a less costly device). As IBM's approach to automating the management of storage resources, it consists of these five components:

- DFSMSdfp (Data Facility Product), the biggest and only non-optional component on this list, provides the data, storage, program, and device management. It includes the following subcomponents: Extended Remote Copy (XRC), Peer-to-Peer Remote Copy (PPRC), Network File System (NFS), FlashCopy, SnapShot, and Concurrent Copy.
- DFSMSdss (Data Set Services) is the high-speed data mover used, among other things, to copy data from disk to disk or disk to tape.
- DFSMSHsm (Hierarchical Storage Manager) quickly and automatically assists in moving data to less costly devices.
- DFSMSrmm (Removable Media Manager) helps deal with removable media.
- DFSMSStvs (Transactional VSAM Services) covers data integrity for shared VSAM by batch and CICS.

With DB2 10, SMS will manage and control all new catalog changes, all the conversions to universal table spaces and all new catalog and directory indexes and table spaces. If you've been putting off taking advantage of DFSMS with DB2, it's time to get comfortable with it. All the new indexes and table spaces being added in DB2 10 will also use Extended Addressability (EA). If DB2 detects that the catalog and directory aren't in an SMS-compliant environment, an abend will occur.

The migration job, DSNTIJSS, provides the necessary DFSMS classes for customers currently not using DFSMS. The environment created by DSNTIJSS is only for the DB2 catalog and DB2 directory objects. No other DB2 data sets are covered. DSNTIJSS must be completed before the installation/migration jobs, DSNTIJTC or DSNTIJEN, are executed.

Start working with your storage management group, or at least warn them about what's just over the horizon. If you've been dodging using DFSMS for all these years, dodge no longer. Learn more in these publications:

- *z/OS V1R10.0-V1R11.0 DFSMS Introduction*
- *z/OS V1R11.0 DFSMS Implementing System-Managed Storage*
- *z/OS V1R12.0 DFSMSHsm Storage Administration Guide*
- *z/OS V1R12.0 DFSMSdfp Storage Administration Reference*
- *z/OS V1R12.0 DFSMSdss Storage Administration Guide*

Migration Steps

DB2 10 doesn't force you to move to DFSMS alone. When installing (or migrating) to DB2 10, the installation panels will help you specify the correct SMS information DB2 10 needs for the catalog, directory, and their indexes. DB2 10 provides entries on the installation/migration panel, DSNTIPA2, for specifying the SMS data classes, management classes, and storage classes for the catalog, directory, and catalog/directory indexes similar to what's currently available on the DB2 9 panel, DSNTIPA3. However, unlike DB2 9, DB2 10 also provides three new DSNZPARM parameters for each catalog and directory (CATDDACL, CATDMGCL, and CATDSTCL on the DSN6SPRM macro), and the catalog and directory indexes (CATXDAACL, CATXMGCL, and CATXSTCL also on the DSN6SPRM macro).

Before populating the fields on the installation/migration panel, your storage administrator should provide you with the SMS data classes, management classes, storage classes, and any necessary SMS Automatic Class Selection (ACS) routines. Make sure SMS is configured to allocate all the DB2 catalog and directory data sets in extended format and use extended addressability.

DSNTIJTC migrates the DB2 catalog and directory from DB2 9 or Version 8 to DB2. It will also create any new catalog objects (new directory objects are created by DSNTIJEN), add columns to existing catalog tables, and create and update indexes on the catalog tables for DB2 10. DSNTIJTC creates the IBM-supplied indexes. These changes could require some of the IBM-supplied catalog indexes to be rebuilt to remove the advisory rebuild-pending status. Change data capture is also disabled as part of the migration to DB2 10 CM. If you plan to use change data capture with SYSROUTINES catalog table, it must be enabled. All the aforementioned catalog and directory changes are completed before or as part of the upgrade to DB2 CM.

DB2 Catalog Links

DB2 10 is making significant changes to the DB2 catalog and directory. To appreciate where the catalog is heading, consider where it is today by looking at DB2 catalog links. This feature is best explained with an example using catalog table space SYSDBASE.

—

—

Before DB2 10, the catalog table space SYSDBASE contained the catalog tables SYSCOLAUTH, SYSCOLUMNS, SYSFIELDS, SYSFORIGNKEYS, SYSINDEXES, SYSINDEXPART, SYSKEYS, SYSRELS, SYSSYNONYMS, SYSTABAUTH, SYSTABLEPART, SYSTABLES, and SYSTABLESPACE. This list of catalog tables covers just about every object that could be created in a database after successfully executing a CREATE DATABASE statement.

You can examine these tables with DSN1PRNT, which exposes much more information about a row in one of the tables in the previous paragraph than the columns listed in the SQL reference manual's catalog description. Using SYSTABLES as an example, if a SELECT * were run against SYSTABLES, the first column displayed would be the column containing the table name, the column "NAME." However, in the DSN1PRNT output for a data page in a SYSDBASE containing a SYSTABLES' row, besides the usual row header information, you would find:

- A 4-byte forward ring pointer pointing to the next SYSTABLES record or SYSTABLESPACE record
- Followed by a 4-byte backward ring pointer pointing to the previous SYSTABLES record or SYSTABLESPACE record
- A 4-byte child pointer pointing to a SYSCOLUMNS record
- A 4-byte child pointer pointing to a SYSRELS record
- A 4-byte child pointer pointing to a SYSINDEXES record
- A 4-byte child pointer pointing to a SYSTABAUTH record
- A 4-byte child pointer pointing to a SYSSYNONYMS record.

All this information appears before any data that might be displayed via an SQL SELECT statement or any column listed in the SQL reference manual. It's information exposed only by running something such as DSN1PRNT.

What should be apparent, based on the list of catalog tables contained in SYSDBASE, is the tremendous potential for contention on catalog table spaces with multiple tables. After all, SYSDBASE does use page-level locking. Any single data page in SYSDBASE could contain rows from one of the multiple tables that comprise SYSDBASE. Inserting a row for a new table into SYSTABLES, for example, could inadvertently lock a table space page containing rows for multiple other catalog tables, preventing someone else from completing their ALTER or CREATE.

DB2 10 to the Rescue

DB2 10 removes the links from the SYSDBASE, SYSPLAN, SYSDBAUT, SYSVIEWS, and SYSGROUP table spaces in the DB2 catalog (DSNDB06) and the DBD01 table space in the DB2

directory (DSNDB01). The six table spaces will be completely removed from DB2 with each table in those disappearing table spaces being moved to their own table space—one table per table space just as we suggest for user data. That translates to 13 new table spaces in the catalog database DSNDB06 just to replace the one table space, SYSDBASE. An example would be SYSTABLES moving to the new table space, SYSTSTAB. Of course, the huge benefit of this one-to-one move is that, in the case of SYSTABLES, as an example, SYSTABLES is now the only table in the table space, SYSTSTAB. This should have a huge positive effect on reducing catalog contention.

Another significant benefit is that the new table spaces being created will all be partition-by-growth universal table spaces; they'll use the new (as of DB2 9) Reordered Row Format (RRF) and leverage row-level locking. When they're migrated to partition-by-growth table spaces, they'll use a SEGSIZE 64 and MAXPARTITIONS set to 1.

Besides the five table spaces that use links, two other table spaces in DSNDB06 are being replaced: SYSOBJ and SYSPKAGE. These seven pre-DB2 10 NFM table spaces will be replaced by a total of 53 new table spaces, each containing one and only one table. The complete list of new table spaces will be available in Appendix A of the *DB2 10 SQL Reference*.

All these catalog changes are performed by job DSNTIJEN as part of the move to DB2 10 ENFM. DSNTIJEN is comprised of multiple steps that use CATENFM and REORG SHRLEVEL REFERENCE to move the catalog from shared table spaces to dedicated table spaces.

If you're upgrading from DB2 V8 to DB2 10 using skip migration, the DB2 catalog tables SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS are created and loaded from the original Real-Time Statistics tables (RTSs) SYSIBM.TABLESPACESTATS and SYSIBM.INDEXSPACESTATS. Because DB2 V8 didn't support BIGINT, TOTALROWS and SPACE in SYSTABLESPACESTATS and TOTALENTRIES in SYSINDEXSPACESTATS are also converted from FLOAT to BIGINT. DSNTIJEN is also responsible for changing the data types and lengths of some other existing catalog columns.

Other Changes

Besides moving the existing catalog tables to the partition-by-growth universal table spaces, with each table ending up in its own table space, DB2 10 introduces a few other changes:

- The directory objects (SYSUTIL, SYSUTILX, and SYSLGRNX) were changed to row-level locking.
- The columns in SYSUTIL and SYSUTILX were reordered, and their page sizes changed to 32KB.
- SELECT is now allowed against SYSLGRNX.

In addition, the directory table space, SPT01, will move to a partition-by-growth table space with a page size of 32KB. SPT01 will have a maximum partition size of 64GB and the maximum number of partitions will be set to 1. SPT01 will also take advantage of two Binary Large Object (BLOB) columns to eliminate the size issues found in previous versions of DB2; one BLOB column will be for the package section data and the other for the static explain blocks. Compression introduced in DB2 9 will still work for the base table rows of SPT01.

Finally, the user-defined tables SYSIBM.SYSDUMMYA, SYSIBM.SYSDUMMYE, and SYSIBM.SYSDUMMYU are moved into the DB2 catalog as part of the migration to CM. The columns SYSVIEWS.TEXT, SYSTRIGGERS.TEXT, SYSROUTINES_SRC.CREATEMST, and SYSPACKSTMT.TEXT are being moved from text columns to LOB columns with DB2 10 NFM.

Conclusion

The catalog and directory are the heart and soul of DB2—the parts that keep everything tied together and working correctly. They're the single most important external components of DB2. The extensive enhancements to the catalog and directory reflect their importance. These are exciting changes you should find beneficial.

Acknowledgement: Special thanks to Chris Leung, Frank Bower and Jay Yothers, DB2 for z/OS Development, IBM Silicon Valley Lab, for their help with this article.

To see the sidebar "Skip-Level Migration", click [here](#).

—
—