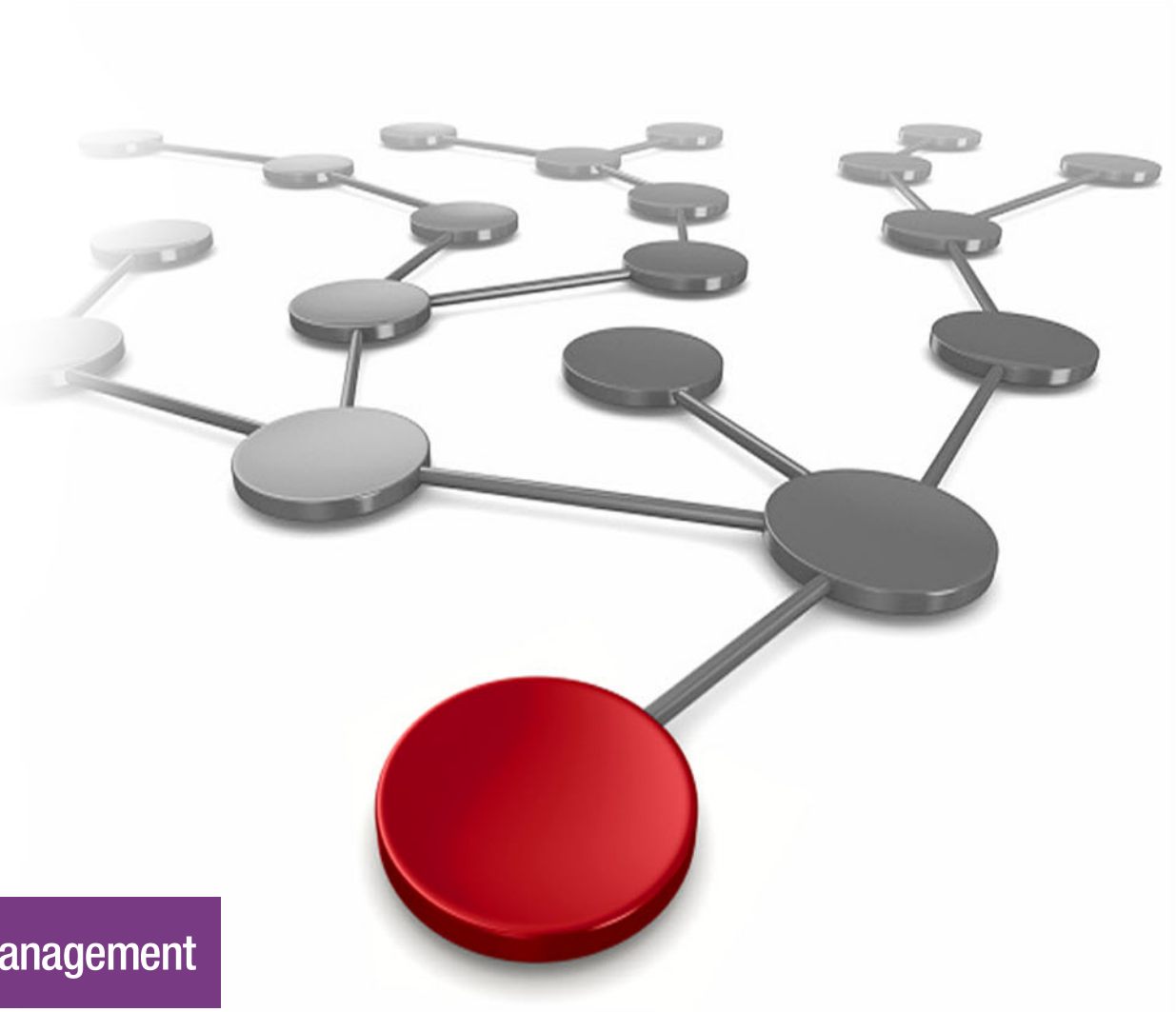
Redbooks
ibm.com/redbooks

Db2 Skeleton Cloning
Protecting Your Production Environment

Paul Bartak

Tim Willging



Information Management

IBM

Redpaper

Overview

IBM® Db2® for z/OS® is well known as the gold-standard information steward. Deep synergy with the z/OS operating system and System Z platform provides support for the highest transaction volumes with the ultimate levels of availability. Just like any high-performance engine, occasional maintenance or upgrades are needed to maintain peak performance and to incorporate new features.

Those that demand the highest standards and protection of their production environments know that you want to test changes outside of production first. It is common to have development or test environments for application development and verification. What about applying Db2 maintenance or performing migrations to new version or release levels? Sure, you probably perform these activities outside of production first, but are these environments similar enough to production to surface the same results as those you might encounter in production? Your production Db2 Catalog & Directory often has a different mix and complexity of objects, which were created at different levels of Db2, that can span decades of time.

The best test of these activities is against your production system, but this is the system that we want to protect. How can we accomplish this? Clone it! Skeleton cloning produces a specific kind of clone, which provides a replica of the portions of your Db2 production environment that are needed to complete your testing. You can use the skeleton clone to find issues before they occur in production. This process allows you to refine maintenance steps in a safe environment and to minimize potential downtime when performing the same steps in a production system.

This IBM Redpaper™ publication gives a high-level overview of the IBM Db2 Cloning Tool and includes specific use cases for the tool. It also details the skeleton cloning process, which you can use to test migration, function levels, and maintenance, and includes demo examples that show a Db2 11 to 12 migration test using skeleton cloning.

It includes the following main topics:

- IBM Db2 Cloning Tool overview
- Skeleton cloning overview
- Skeleton cloning process
- Determining the source domain
- Volume-level cloning
- Data set-level cloning
- Comparing volume-level cloning to data set-level cloning
- Db2 migration testing
- Skeleton cloning clean up
- Summary

IBM Db2 Cloning Tool overview

Skeleton cloning is a subset of features provided by the IBM Db2 Cloning Tool. Before we dive into the details of skeleton cloning, let's review the full scope of the Cloning Tool capabilities.

The Cloning Tool is a purpose-built tool for the rapid cloning of Db2 subsystems or data sharing groups, or subsets of those subsystems or data sharing groups. It performs all necessary processes to clone Db2 objects and data, and it is also storage aware. Unlike recovery products that extend their functionality into the cloning space or manual methods, such as Unload and Load, the cloning tool focuses on the fastest methods to perform discovery, replication, and conditioning of your target system for rapid provisioning.

Whenever possible, the Cloning Tool invokes the fast replication services of your storage subsystem for the most efficient creation of the target environment. It can also use image copies as a source. The rapid, efficient cloning processes support multiple use case for production-like replicas without production impact, such as:

- Creating or refreshing of robust, life-like quality assurance (QA) or test environments
- Moving a workload to another environment, such as query environments
- Separating forensics, auditing, or analytics from production
- Testing migrations or maintenance
- Provisioning in support of agile development

The Cloning Tool is full featured. In one product, you'll find the following capabilities:

- *Subsystem cloning*, which includes full data sharing and skeleton cloning support. These clones can be created in the same or different logical partitions (LPARs). What differentiates a subsystem clone is that the source Db2 catalog, directory, bootstrap data sets, and active logs (optionally the archive logs can be included) are cloned to a different subsystem or data sharing group name. Optionally, if the source is a data sharing group, the target can have the same or fewer members or can be converted to non-data sharing. System Level Backups can serve as the source to a subsystem clone.
- *Application cloning*, which provides the capability to clone a subset of a Db2 subsystem or data sharing group. A source set of databases, table spaces, index spaces, tables, indexes, or storage groups can be identified for cloning. This list can be expanded with some simple options to include referentially related tables, associated indexes, and auxiliary objects. The target for this clone can be the same or a different Db2 and the same or a different LPAR. Flexible object renaming and Data Definition Language (DDL) attribute changes allow you to customize the target objects. Missing target databases, tables spaces, tables, index spaces, indexes, and auxiliary objects are created using source object definitions and translation rules for names and attributes. Log Apply of source log records to the target objects allow for the source system to remain available during cloning. Optionally, you can choose image copies as the source including IBM FlashCopy® consistent image copies. Basic data masking is offered as an option to apply one of 10 different built-in algorithms at the column level, or a user exit can be chosen to invoke external masking logic provided by the customer.

You can use the Cloning Tool to protect your production environment, while also providing robust production-like environments that are needed for provisioning development, workload isolation, pre-production testing, and support for Db2 continuous delivery.

The remainder of this paper explains the specifics of using skeleton cloning capabilities.

Skeleton cloning overview

A *skeleton clone* is a subset of subsystem cloning. The subsystem cloning process is directed to take portions of the Db2 subsystem or data sharing group, as illustrated in Figure 1.

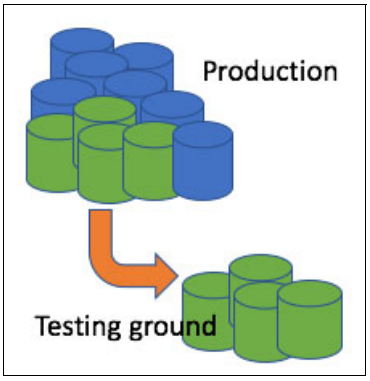


Figure 1 Creating a skeleton clone

A skeleton clone for full migration, function level, or maintenance testing includes the following important artifacts:

- Db2 directory (`DSNDB01`)
- Db2 catalog (`DSNDB06`)
- Bootstrap data sets (BSDS), copy 1 and 2
- Active logs, copy 1 and 2
- Work file data sets (or a plan to re-create them)
- Catalog extensions, which includes objects for the following components:
  - Administrative Scheduler
  - Db2 Analytics Accelerator
  - Java Database Connectivity (JDBC) and Structured Query Language for Java (SQLJ)
  - Plan Program Authorization
  - MQI-based function policies
  - Data Definition Control support
  - Resource limit facility (RLF)
  - XML Schema Repository
  - JSON support
  - REST Services
  - Text Search Administration database
  - Tools database

Optionally, you can include archive logs and selected application spaces in support of customer-specific test plans. You can also use the Application Cloning features of the Cloning Tool at a later time to add selected objects, as needed, to the skeleton clone.

This clone includes the following benefits:

- Gaining confidence for a smooth production change by testing in a production-like clone while protecting your production environment.
- Determining timing estimates for your production change window.
- Early identification and resolution of potential incompatibilities.

- Testing changes that are introduced by a migration or function level, such as:
  - BSDS format, logging, relative byte address (RBA), and log record sequence number (LRSN)
  - Table space changes
  - Other new features

Skeleton cloning can be accomplished using one of the following basic approaches, which are discussed further in this paper:

- *Volume-level copies* are likely to be easier to set up and faster to run but might require more target disk storage. This type of copy can be executed while maintaining source, or production, system availability.
- *Data set-level copies* might take slightly longer to set up, verify, and execute but are likely to require less target disk storage. The source production system must be stopped during the copy, and the target data sets require storage management subsystem (SMS) support for proper data set placement.

Skeleton cloning process

The skeleton cloning process includes the following phases:

- Preparation
- Cloning
- Testing
- Clean up

Preparation

As with any subsystem cloning process, the target Db2 environment must be established within the target z/OS LPAR, which includes the following components:

- The subsystem or subsystems and the group, and the started tasks definitions within z/OS
- Appropriate security definitions
- Target libraries for the current release or function level and the +1 release or function level
- The Cloning Tool special ZParms that are used during target system conditioning. For more information about these special ZParms refer to *DB2 Cloning Tool for z/OS User's Guide*, SC27-6556-01, "Chapter 9: Cloning DB2 systems".

In this phase, you also determine the portions of the source system that are to be copied. This process is different depending on whether you plan to use volume-level copies or data set-level copies. See "Determining the source domain" on page 5 for more information about choosing all the needed volumes or data sets for a proper skeleton clone.

Cloning

This phase creates the skeleton clone and includes the execution of the steps that are needed to perform volume-level or data set-level cloning. When established, cloning is easily repeated to refresh data or to perform repetitive tests.

Testing

Perform your tests against the clone to build confidence that these same procedures will be successful in production. For the purposes of this paper, we explore the Db2 11 to Db2 12 migration steps. However, this process can apply to the function level activations that are associated with Db2 continuous delivery, system maintenance, or any other change to production that needs scrutiny to avoid production issues.

Clean up

Chances are that these changes to production are not a one-time event. Proper cleanup can assist in establishing the clone for the next round of changes, whether those changes occur next month, next quarter, or years from now.

Determining the source domain

To determine the source domain, you need to find all the volumes or data sets that cover the following core artifacts for a skeleton clone:

- Db2 directory (`DSNDB01`)
- Db2 catalog (`DSNDB06`)
- Bootstrap data set (BSDS), copy 1 and 2
- Active logs, copy 1 and 2
- Work file data sets (or plan to re-create them)
- Catalog extensions (described in detail in “Skeleton cloning overview” on page 3)

First, we will look at collecting this information for data set-level cloning, because this same technique can be used to determine the volumes needed for volume-level cloning.

The Cloning Tool **`COPY-BY-DS`** command uses `RENAME-MASK` statements to identify the data sets to be cloned. A sample **`COPY-BY-DS`** job is provided in the `SCKZJCL` product library in the `CKZCPYDS` member. Customize this job for your installation. You can find more information about this command in *DB2 Cloning Tool for z/OS User’s Guide*, SC27-6556-01, “Chapter 24: Db2 Cloning Tool Subsystem Cloning commands”.

Example 1 includes a sample `COPY-BY-DS` statement that clones data sharing group `IDS2` (members `I9A2` and `I9B2`) to data sharing group `ZDS2` (members `Z9A2` and `Z9B2`).

Example 1 Sample COPY-BY-DS statement

```
COPY-BY-DS                                                  -
      REPLACE-UNCONDITIONAL                                     -
      OFFLINE                                                   -
      RENAME-MASKS(                                             -
        I9A2LOG.BSDS0%.DATA          Z9A2LOG.BSDS0%.DATA            -
        I9B2LOG.BSDS0%.DATA          Z9B2LOG.BSDS0%.DATA            -
        I9A2LOG.LOGCOPY%.DS0%.DATA   Z9A2LOG.LOGCOPY%.DS0%.DATA     -
        I9B2LOG.LOGCOPY%.DS0%.DATA   Z9B2LOG.LOGCOPY%.DS0%.DATA     -
        IDS2.DSNDBD.DSNDB01.**       ZDS2.DSNDBD.DSNDB01.**         -
        IDS2.DSNDBD.DSNDB06.**       ZDS2.DSNDBD.DSNDB06.**         -
        IDS2.DSNDBD.WRK*.**          ZDS2.DSNDBD.WRK*.**            -
        IDS2.DSNDBD.DSNACCEL.**      ZDS2.DSNDBD.DSNACCEL.**     -
        IDS2.DSNDBD.DSNADMDB.**      ZDS2.DSNDBD.DSNADMDB.**     -
        IDS2.DSNDBD.DSNATPDB.**      ZDS2.DSNDBD.DSNATPDB.**     -
        IDS2.DSNDBD.DSNHMDB.**       ZDS2.DSNDBD.DSNHMDB.**      -
        IDS2.DSNDBD.DSNMDCDB.**      ZDS2.DSNDBD.DSNMDCDB.**     -
        IDS2.DSNDBD.DSNMQDB.**       ZDS2.DSNDBD.DSNMQDB.**      -
```

```
    IDS2.DSNDBD.DSNRGFDB.**      ZDS2.DSNDBD.DSNRGFDB.**     -
    IDS2.DSNDBD.DSNRLST.**       ZDS2.DSNDBD.DSNRLST.**      -
    IDS2.DSNDBD.DSNXSR.**        ZDS2.DSNDBD.DSNXSR.**       -
    IDS2.DSNDBD.DSN5JSDB.**      ZDS2.DSNDBD.DSN5JSDB.**     -
    IDS2.DSNDBD.DSNSVCDB.**      ZDS2.DSNDBD.DSNSVCDB.**     -
    IDS2.DSNDBD.SYS*.**          ZDS2.DSNDBD.SYS*.**         -
  )                                                        -
  JOURNAL-DDN( JOURNAL ) SIMULATE
```

This example includes statements for the following artifacts:

- The BSDSs
- The active logs
- The Db2 catalog and directory
- The work file data sets (if non-data sharing, replace this with `DSNDB07`)
- Data sets that support the Catalog extensions and Tools objects, which are needed to fully test migration jobs `DSNTIJSG` (commented DDL) and `DSNTIJRT` (Db2 Supplied Routines)

The specific nature of this list eliminates the inclusion of implicit and temporary (`DSNDB04`) databases, because these databases are considered application data sets at most installations. However, it will need to be monitored as new Catalog Extension databases are added over time.

The `SIMULATE` keyword provides reporting without performing the copies. Review the reporting in this job output for message `CKZ06536I Source volumes written to Journal file`. Include these volumes if you plan to perform volume-level cloning.

The `CKZ06516I` message lists the data sets mapped by the `RENAME-MASK` statements, as shown in Figure 2. Verify that the correct data sets are found and mapped to the correct target data sets.

```
CKZ06536I Source volumes written to Journal file: ** SIMULATION **
     DBTL01
     DBTL02
     DBTD04
     DBTD01
     DBTD00
     DBTD02
     DBTD05
     DBTD03
CKZ06516I    Source data sets to be copied, on volumes            Target data sets to be created              Target storage group

     1   1 D I9A2LOG.BSDS01.DATA                          DBTL01  Z9A2LOG.BSDS01.DATA
     1   1 D I9A2LOG.BSDS02.DATA                          DBTL02  Z9A2LOG.BSDS02.DATA
     1   2 D I9B2LOG.BSDS01.DATA                          DBTL01  Z9B2LOG.BSDS01.DATA
     1   2 D I9B2LOG.BSDS02.DATA                          DBTL02  Z9B2LOG.BSDS02.DATA
     1   3 D I9A2LOG.LOGCOPY1.DS01.DATA                   DBTL01  Z9A2LOG.LOGCOPY1.DS01.DATA
     1   3 D I9A2LOG.LOGCOPY1.DS02.DATA                   DBTL01  Z9A2LOG.LOGCOPY1.DS02.DATA
     1   3 D I9A2LOG.LOGCOPY1.DS03.DATA                   DBTL01  Z9A2LOG.LOGCOPY1.DS03.DATA
     1   3 D I9A2LOG.LOGCOPY2.DS01.DATA                   DBTL02  Z9A2LOG.LOGCOPY2.DS01.DATA
     1   3 D I9A2LOG.LOGCOPY2.DS02.DATA                   DBTL02  Z9A2LOG.LOGCOPY2.DS02.DATA
     1   3 D I9A2LOG.LOGCOPY2.DS03.DATA                   DBTL02  Z9A2LOG.LOGCOPY2.DS03.DATA
     1   4 D I9B2LOG.LOGCOPY1.DS01.DATA                   DBTL01  Z9B2LOG.LOGCOPY1.DS01.DATA
     1   4 D I9B2LOG.LOGCOPY1.DS02.DATA                   DBTL01  Z9B2LOG.LOGCOPY1.DS02.DATA
     1   4 D I9B2LOG.LOGCOPY1.DS03.DATA                   DBTL01  Z9B2LOG.LOGCOPY1.DS03.DATA
     1   4 D I9B2LOG.LOGCOPY2.DS01.DATA                   DBTL02  Z9B2LOG.LOGCOPY2.DS01.DATA
     1   4 D I9B2LOG.LOGCOPY2.DS02.DATA                   DBTL02  Z9B2LOG.LOGCOPY2.DS02.DATA
     1   4 D I9B2LOG.LOGCOPY2.DS03.DATA                   DBTL02  Z9B2LOG.LOGCOPY2.DS03.DATA
     1   5 D IDS2.DSNDBD.DSNDB01.DBD01.I0001.A001         DBTD04  ZDS2.DSNDBD.DSNDB01.DBD01.I0001.A001
     1   5 D IDS2.DSNDBD.DSNDB01.DSNDB01X.I0001.A001      DBTD01  ZDS2.DSNDBD.DSNDB01.DSNDB01X.I0001.A001
     1   5 D IDS2.DSNDBD.DSNDB01.DSNDB1XA.I0001.A001      DBTD01  ZDS2.DSNDBD.DSNDB01.DSNDB1XA.I0001.A001
```

Figure 2 Results of the CKZ06516I message

You might have your own internal procedure to determine a complete set of volumes for your skeleton clone. An alternative approach is provided here for your consideration. A sample REXX routine can invoke Interactive System Productivity Facility (ISPF) services to list data

sets that meet a few wild card masks. This output is then consumed by a spreadsheet to consolidate the volumes that are needed for skeleton cloning objects and to eliminate implicit and temporary databases. This approach automatically finds new data sets that meet the DSN*, WRK*, and SYS* masks. If you are interested in investigating this approach, you can find Skeleton Cloning samples online.

Volume-level cloning

Regardless of the method that you use (volume-level or data set-level), follow the instructions for volume-level cloning to generate a base set of jobs.

With the source volumes identified, unless you've purposely segregated these skeleton artifacts to their own volumes, it is likely that more data sets than are needed will be copied. However, an IBM FlashCopy can be accomplished in seconds, even with the additional data sets that will not be used. If needed, you can consolidate data sets after the cloning steps to reduce the storage foot print.

Make sure that you have a target volume for each source volume. The target volumes do not have to be empty, but they are about to be overwritten. IBM z/OS catalog data sets cannot exist on the target volumes, because if they do, the copy process will stop.

You can use the Cloning Tool ISPF panels to set up profiles with the rules, parameters, and definitions to build the jobs that are needed for this operation. Choose the **Subsystem Cloning** option to begin building the profile (shown in Figure 3 on page 8). There are 12 unique jobs created, some of which are repeated when cloning a data sharing group for all the members in the group (starts, stops, and BSDS updates). Some jobs are run only if problems are encountered during the conditioning of the target. For skeleton cloning, nine of the created jobs are needed to complete the clone. In this case, three of them are added for a second member of a data sharing group. After these jobs are created and tested, the jobs can be run repeatedly to rapidly re-create a skeleton clone. The profile or the jobs can be copied to set up other skeleton cloning scenarios.

You have the following options to make sure the volumes are consistent during the copy operation:

- Offline
  - Stop the source Db2 during the copy.
- Online
  - Issue the **–DB2 SET LOG SUSPEND/RESUME** commands around the copy.
  - Choose the FlashCopy Consistent option (no **LOG SUSPEND** required).

Volume-level skeleton cloning profile details

As required by any subsystem cloning process, make sure that the z/OS environment is set up for the target, including high-level qualifiers, subsystem identifiers (SSIDs), started tasks, target libraries, security, and `DSNTIJUZ` assembled modules (including the special ZParms mentioned previously) are in place.

```
                              Edit DB2 Subsystem Clone Profile
Option ===>

Creator  . . . : TS5858       Name  . . . . : IDS2 2 ZDS2 SKELETON CLONE(VOL
Share Option . : VIEW ONLY    Description . . Volume level skeleton clone V1  >

1  Select Source and Target DB2 subsystems
2  Select Source and Target Volume Pairing
3  Select Source and Target ICF catalogs
4  Select Rename masks
5  Select other parameters
```

Figure 3 Subsystem Cloning Profile main panel

Follow these steps:

1. Verify that the source and target DB2 subsystem information is set up with the following information:
   - High-level qualifiers for the catalog, directory, BSDSs, and active logs
   - Target DDF information
   - Ensure the `DB2UPDATE` option has `DB2-XCFCLEAN=YES`
   - Ensure the `DB2UTILXCLEAN` option has `SYSUTILX=YES`
2. Enter the source and target volume per the previous analysis.
3. Define the source and target integrated catalog facility (ICF) catalog data set pairs.
4. Define the Rename masks for the source and target data sets. Include a source mask of `CATALOG.**` to a target mask of `GARBAGE.**` to cover any ICF data sets on source volumes (Figure 4). Also, define exclude masks for any source data sets on source volumes that are not covered by the Rename masks and that are not wanted for your target clone.

```
                                                          Rename Masks
Command ===>

Commands:       A - Add Line
Line commands: D - Delete Line

Creator  . . . : TS5858       Name  . . . . : IDS2 2 ZDS2 SKELETON CLONE(VOL
Share Option . : VIEW ONLY    Description . : Volume level skeleton clone V1>


Cmd Source Mask                          Target Mask
--- ----------------------------------> ---------------------------------->

    CATALOG.**                           GARBAGE.**
```

Figure 4 Rename Masks

5. For other parameters, ensure that the following options are set appropriately:
   - For the **`COPY`** command, set the following `DATA-MOVER` parameters:
     - `CONSISTENT=YES` for online copy without a log suspend/resume.
     - `FASTREP=REQ` runs only if Fast Replication is available. Use `FASTREP=PREF` if you want to fall back to host based copies, which might take significantly longer.
   - For the **`RENAME`** command, set the following parameters:
     - Use `RECATALOG=YES` if there are existing data sets with the target names.
     - Set `RENAME TYPE=SAFE` and `VOLBKUP-DDN` to a `ddname` of choice to be able to restart the process at the **`RENAME`** step.

– Review the SMS Class settings, if applicable.

– To refresh the skeleton clone when testing is completed, set up the following options, which generate additional jobs for clean up:

  - Set **BCSCLEAN**, CLEANUP-CATALOG-ORPHANS=YES.
  - Define the CLEANUP-CATALOG-DSNMASKS values.

– Build the subsystem cloning jobs.

Table 1 lists the jobs created from the Cloning Tool user interface build process, as members in a JCL library. The table also includes a description and notes or instructions, along with elapsed times for running the jobs in a lab demo environment.

Table 1 Volume-level Skeleton Cloning jobs

Generated job prefix	Description	Instructions / notes	Demo timing
ST01	SET LOG SUSPEND	Not needed with Consistency Groups.	
ST02	Copy Volumes	Required 8 volumes copied in the demo test. Might need to “move” volumes to the target LPAR.	0:00:07
ST03	SET LOG RESUME	Not needed with Consistency Groups.	
ST04	Copy Check job	Not needed	
ST05	RENAME	Required SAFE mode includes volume backups.	0:00:38
ST06/7	Update target BSDSs	Required Demo included 2 members. Additional members can change the number used in the generated job.	0:00:12
ST08/9	Start 1 target member. Start / Stop other target members.	Required 2 members for the demo. 1 member left running on the Cloning Tool special ZParms.	0:01:23
ST10	Fix the target Db2 Catalog and Directory status	Required	0:00:02
ST11/12/13	Fix the target directory if ST10 fails	Not needed	
ST14	Update target Catalog	Required Updates VCAT, VOLUME, GROUPMBR.	0:00:04
ST15	Fix target application databases	Not needed unless application spaces are included in the skeleton clone.	
ST16	Stop the running target member	Required	0:00:26
ST20	Target SYSUTILX Clean Up	Recommended	0:00:01
ST21	Start target Db2/members on regular ZParms	Required	0:00:30
Total Volume Cloning Time			0:03:23

In this demo environment, the eight volumes resulted in approximately 128,000 data sets cloned, totaling about 12.5 GB.

Data set-level cloning

The **COPY-BY-DS** command is used to create this clone. Currently, this command is not supported in the Cloning Tool user interface. Therefore, you need to complete the process described in "Volume-level cloning" on page 7 to create a base set of jobs. Then, replace the COPY job (ST05) in the volume-level demo example, with the **COPY-BY-DS** job using the set up as described in "Determining the source domain" on page 5. Only the required or recommended jobs listed in Table 1 on page 9 will be used.

Note: Remember to remove the SIMULATE keyword to perform the copying of data sets.

This method requires you to stop the source system during the copy. Therefore, the ST01 job is modified to issue the **DB2STOP** command, and the ST03 job is modified to issue the **DB2START** command. Table 2 lists the jobs that are modified from the previous volume-level skeleton clone profile and is limited to the jobs that were used to create the clone. As before, the timings represent elapsed times collected in a lab demo environment.

Table 2 Data set-level Skeleton Cloning jobs

Generated job prefix	Description	Instructions / notes	Demo timing
ST02	STOP DB2	Required for data set-level. Stopping the source Db2/members. 2 members stopped.	0:01:10
ST02	Copy source data sets	Required COPY-BY-DS command performs the copy and rename in a single step, therefore there will not be a later RENAME job.	0:02:13
ST03	START DB2	Required Start the source Db2/members.	0:00:23
ST04	Update the target BSDSs	Required	0:00:06
ST05	Start the target Db2/members	Required Only one member remains running. Started with special ZParms.	0:00:50
ST06	Resolve target catalog/directory statuses	Required	0:00:03
ST07	Update Catalog	Required Updates VCAT, VOLUME, GROUPMBR.	0:00:03
ST08	Stop the running member	Required	0:00:25
ST09	SYSUTILX Clean Up	Recommended	0:00:01
ST21	Start Db2/members on normal ZParms	Required	0:00:23
Total Data Set Cloning Time			0:05:37

In this demo environment, the **COPY-BY-DS** command resulted in 600 data sets cloned, totaling about 3.5 GB.

Comparing volume-level cloning to data set-level cloning

Table 3 compares the volume-level cloning results to the data set-level cloning results in the demo environment.

Table 3 Comparison of volume-level and data set-level information

	Volume-level cloning	**Data set-level cloning**
Volumes	8 source volumes & 8 target volumes	8 source volumes and 9 target volumes Target placement is SMS managed.
Data sets clone (approximate)	128K	600
Total target clone space (approximate)	12.5 GB	3.5 GB
Demo combined elapsed time	0:03:23	0:05:37
Considerations	Consistency Group keeps the source available without the need for a log suspension	The source must be stopped during the copy and SMS is needed for data set placement.

Db2 migration testing

There are 31 documented migration steps for Db2 12. Several of these steps involve checking specific items in the current version (V11) catalog. Some of these steps are not provided as a `DSNTIxxx` job and require the customer to generate them or to obtain SQL from a sample or documentation.

The following steps were successfully tested against the skeleton clone but are not included in the repeatable, timed steps:

- `DSNTIJPM` (`DSNTIJPx` in the current version, `DSNTIJPC` for V11)
- `DSN1COPY` with `CHECK` and `CHECK INDEX` for the catalog and directory
- Determining invalid packages
- `DSNTESQ` (catalog consistency)
- `DSNTIJIC` (catalog and directory backup)
- `DSNTIJCI` (catalog integrity check once at the V12 level)
- `DSNTIJRV` (to verify routines)

Other steps involve set up outside of the migration process and typically are already completed in customer environments or do not apply. These steps were not run:

- `DSNTIJVC` (connect to TSO), similar steps for IBM IMS™ and IBM CICS®
- `DSNTIJIN` (defining subsystem data sets)
- `DSNTIJEX` (define authorization exits)
- IPL for ERLY code
- V11 or V12 installation verification procedures (IVPs)
- `DSNTIJNG` (application programming module to `SQLLEVEL V12R1M100`)

Fallback and remigration were not tested, including `DSNTIJNH` (halts `DSNTIJTC`). Table 4 shows the timing for the migration jobs.

Table 4 Migration job timing

Migration job or step	Description	Instructions / notes	Demo timing
	ZParm changes	Done manually. There are several removed parameters, including ALCUNIT. Adjust PRIQTY & SECQTY to avoid oversized archive logs.	0:10:00
	Stop Db2, update PROCs, Start 1 member	Stop / Start done manually, time estimated. DSNT500I, 00C900A1 messages about DSNDB06 when started on V12.	0:05:00
DSNTIJTC	CATMAINT To CATALOG LEVEL(V12R1M100)	There is no DSNTIJEN for V12. Single step for catalog changes.	0:02:26
DSNTIJTM	Bind dynamic SQL Programs		0:00:08
DSNTIJSG	Bind SPUFI & DCLGEN		0:00:01
DSNTIJRT	Db2 supplied routines	For the M100 level	0:00:34
	Start additional members		
DSNTIJAF	Activate Function Level M500	There's no DSNTIJNF for V12	0:00:00
DSNTIJRT	Db2 supplied routines	For the M500 level	0:00:06
Total Migration Time			0:18:15

Skeleton cloning clean up

You can use the `BCSCLEAN` job from the volume-level cloning steps to uncatalog the cloned data sets. This process is meant to assist with re-cloning of the target when using the volume-level copy method. Data sets are not deleted and are replaced with the next volume-level clone. Consequently, this process does not work well for data set-level cloning clean up or to remove the volume-level clone.

A more thorough approach to cleaning up the clone, created via either method, is to use `IDCAMS DELETE` statements.

When performing skeleton cloning clean up, be sure not to delete the following elements:

- Source data sets
- Target distribution libraries (SDSN*).

Summary

When established, you can use the skeleton cloning process to rapidly build test environments for migration, function level, and maintenance validation. You can even extend the process into application spaces for customized testing within minutes. The provided demo examples in this paper show that a Db2 11 to 12 migration test, with skeleton cloning, was consistently conducted in under 30 minutes. Your production catalog, directory, logs, and so on are likely larger and might take longer. However, with fast replication technology and the Cloning Tool's rapid renaming processes, it is expected that you'll find the process efficient and the extra protection for your production environment as valuable insurance.

There are many other use cases for the IBM Db2 Cloning Tool. If you are an existing Cloning Tool customer, consider skeleton cloning as an additional use case to drive more value from your investment. If you are considering the Cloning Tool for your environment, know that skeleton cloning is only one feature that is available in this robust, purpose-built tool that will assist the roles of systems, database, storage administration, and application development. The Cloning Tool can even offer REST options to assist with provisioning demands.

Authors

This paper was produced by the following:

Paul Bartak has been working with IBM Information Management solutions since 1985, including the first version of DB2 on MVS™. As an IBM customer for 14 years Paul worked in Application Development, Database Administration, and DBA Management - working with IBM & ISV Labs on various projects. With IBM for 18 years his roles included that of a DB2 Advisor, developing the V8 – V11 Migration Planning Workshops, pureXML and Temporal Proof of Technologies; and a Client Technical Leader, working across a large payments processing company and the IBM account team. Paul holds multiple certifications across Information Management and Cloud domains and recently joined Rocket Software as a Principle Solutions Advisor.

Tim Willging joined the Rocket Software team in 2005 and was named a Distinguished Engineer in 2015. With over 25 years developing enterprise software tools, Tim has designed and authored several database products that focus on backup and recovery, cloning, administration, monitoring, and change management. He has recently focused on solutions for advanced analytics on z/OS. He is currently based in Chicago and is an alumnus of Northern Illinois University with an emphasis in theoretical computing.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

REDP-5473-00

ISBN 0738456446

Printed in U.S.A.