

10 Things to expect from a DB2 Cloning Tool

Here is a brief overview of functions that should be provided by a modern DB2 cloning tool. The requirement to copy DB2 data has become an increasing concern in recent years. Whether providing test data for quality assurance or for application development, there is a need for an efficient and reliable tool to do the job. The following list briefly describes features that should be reasonably expected from a cloning tool by most users.

1. Ease of Use

The creation of a new copy process needs to be easy, intuitive and convenient. An ISPF or workstation interface should guide users through the specification tasks needed to set up a data replication. It should ask for important options, source and target DB2 subsystems, and it should query the objects to be copied. A generic selection of databases, tablespaces and tables using wild cards is useful for copying a large number of objects. Additionally, the objects may need to be renamed, for example, when the target DB2 is used as a development system using different schemata and database names.

The process of defining a task should be completed in a couple of minutes. After a task has been created, it must be ready for execution. The jobs required to execute the copy process should be tailored in advance. The cloning tool should also take care of generating and executing the DDL for target objects do not exist yet. When they exist, it is necessary to ensure that the structures match with those of the source. It is not uncommon for developers to add new columns to their data model, and the cloning tool should be able to deal with such situations. It should allow the choice to drop and recreate the objects with the same attributes as the source objects, or to use a fall back mechanism to transfer the data with DB2 utilities like UNLOAD/LOAD.

A cloning tool should be able to transfer the data to the target environment even if the structures do not match. Database administrators rely on automated processes that run every night, and naturally expect them to have executed successfully by next morning. Small structural differences should not cause an entire process to fail. Someone in the management chain always gets a call when teams of developers must sit idle waiting for test data. A cloning tool should notify the staff of any differences found, so an informed decision can be made on whether to drop and recreate the objects in question or not.

2. Efficient Processing and Wise Use of Resources

The volume of data is continuously growing, and a cloning tool needs use the available resources in an efficient way. It needs to shorten processing windows wall clock times and reduce CPU consumption. Ideally, a cloning process should fit nicely into the night shift operations cycle, but with conventional copy methods using UNLOAD/LOAD the jobs simply do not fit into a batch window. This leads to either a subset of the data being copied every night shift, or the full data refreshes are limited to weekend or other less desirable frequencies.

These restrictions should not exist when using a vendor product to copy data on a regular basis. The tool should be ten times faster and use only 10% of the CPU time that UNLOAD/LOAD or similar would require thus opening up new possibilities. Modern cloning tools offer this functionality. They might use hardware-assisted copy facilities like IBM's Flash Copy 2, EMC Time Finder or similar tools. With these tools it is possible to copy a large amount of data in record time; **however**, hardware-assisted copy facilities limit flexibility. A DB2 table space contains internal IDs, a different one for each table. These IDs, called OBIDs, need to be modified, because the target DB2 may have assigned a different OBID to the corresponding target table. It is impossible for a cloning tool that is based on Flash Copy to translate these OBIDs during the copy process itself because hardware based facilities do not allow modifying the data while they are being copied. Instead, it is necessary to change these OBIDs in the target after the data has been transferred. This is usually time consuming and delays the availability of the target environment.

3. Eliminate Manual Intervention

A cloning tool should automate most of the steps needed, so that little or no manual intervention is required from a DBA. The automation starts with the definition of the copy process and ends with the start of the target objects after all data has been copied. There are a few cloning tools on the market that claim to automate all the needed steps, but upon a closer look there are numerous situations where manual intervention is still required. Full automation is almost always limited to situations where your source environment is perfectly clean, freshly reorganized and all objects were created in the current version of DB2. In reality, many objects are migrated from earlier releases of DB2, and for these objects it is probably necessary to do manual corrections. Only BCV5™ covers almost every possible situation, and either executes the required steps automatically or reports errors. Here are just some examples of how BCV5 simplifies the process:

- Integrated check for reorganization.
- Many consistency checks: Restricted states, duplicate names, table and view dependencies.
- Automatic execution of REPAIR VERSIONS if table space or indexes are versioned.
- Automatic rebuild of additional target indexes.
- Automatic fallback to UNLOAD/LOAD if objects cannot be copied on VSAM level.
- Automatic definition of the target page sets with the required size

4. Interface to Other Applications

An ISPF interface is an easy way for new users to define new copy tasks. It can also be used for an ad-hoc definition of copy tasks if an urgent requirement pops up. Imagine the impact of to create dozens or hundreds of different copy tasks. It is not feasible to go through every panel hundreds of times. A professional cloning tool resolves this problem by allowing DBAs to define a copy process either from batch (JCL job) or from their own applications. With a simple description language it is easy to define new copy tasks within minutes. Store a CREATE statement as a template and reuse it every time there is a need to create a new copy task.

```
CREATE COPY TASK "PRCLONE"
FROM "DB8G" ON "PROD"
TO "DB8G" ON "DEV"
COPY_STATISTICS "N"
DROP_OPTION "N"
SELECTION (
  DATABASE "DSN8%")
MAPPING
  OBJECT ("D")
  FIELD ("N")
  TARGET "HUGO%"
MAPPING (
  OBJECT ("%")
  FIELD ("C")
  TARGET "HUGO") 444
```

Simply edit and modify the selection and rename rules of the statement, submit the generated jobs, and the task is done. A batch interface is also quite important for developers and testers, who usually do not have the authorization to carry out cloning operations, to define a copy processes. With a batch interface developers can specify their request and hand it over to the DBAs, which double check the request and submit it for execution at the most convenient time. Also, the data privacy manager can verify that the developer is authorized to copy the requested data and/or that any mandated data masking is being observed.

5. Restartable Process

Murphy's Law has never been kind to IT, and even the best cloning tool will falter from time to time. The target DB2 storage groups might run out of space, authorizations might be missing or target SMS settings improperly configured. It is essential that a solid cloning tool have restart capabilities. This is especially crucial when copying large amounts of data. If the failure occurs near the end of the process a complete rerun might be necessary without this recovery ability. The tool must allow the DBA to fix the problem and resubmit the job, and then only have the outstanding objects copied. This saves a lot of time and avoids unnecessary data movement. The DBA only needs to take care of fixing the underlying cause of the error. After that the cloning tool automatically carries out the remaining work. It knows which objects (tablespaces, indexes) have already been copied and processes only the objects that are missing.

6. Data Privacy Considerations

Privacy is important when copying data from a secure production environment to a testing or QA testbed where it is more vulnerable to prying eyes. Government and company auditors mandate privacy guidelines that often require sensitive data fields to be masked in none production environments. The data privacy manager knows and determines what should be masked, while the application developers know where to find the fields and how a particular

anonymization method impacts the application. For example, if an application expects an e-mail address in a field and the masking method just inserts random characters, the application will probably fail because the field does not contain a valid e-mail address anymore. Masking fields requires an understanding of the data and its uses.

A cloning tool should enforce the masking of selected fields. The reason is not so much that users with a malicious intent can be stopped before they copy data without masking. If they are allowed to read a table, they already have access to all the information. Instead, the purpose is to avoid accidental exposure of sensitive information in non-secured environments. The cloning tool should mask data from a table whenever that table is involved in a cloning process.

The masking process should support all commonly used data types, and it should be able to convert between different data types as required. When specifying rules for the masking process, a standard language like SQL is very beneficial because it is easy to learn and most DBAs are already familiar with it. It also allows defining masking algorithms with DB2 user defined functions. These functions can be written in a language of choice, and they can perform operations that would not be possible with the feature set of normal SQL.

A masking repository is required to control and enforce the anonymization for particular tables. Typically, data is masked only for tables that contain sensitive fields that should be kept private. A central repository stores which columns of which tables always require masking. All users who copy one of these tables will automatically get an anonymized copy in the target without any additional manual intervention.

7. Access to Older Generations

A cloning tool should allow making regular backups of a source environment, and provide an option to restore any generation from the repository of backups at a later point in time. This feature is particularly useful when building education or training environments because all changes that have been made by a class can be reverted with a few simple jobs. Assume that a new release of an application is published and that users must be trained on the new application. A backup of the environment is created and used to train many different groups of users with live data which can be refreshed at will. The restore process is fully automated and runs with negligible resource consumption.

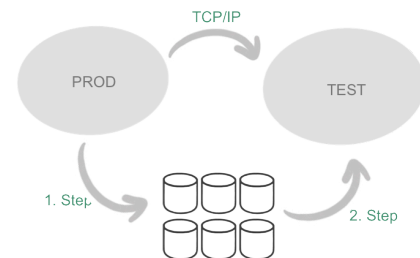
The cloning tools that offer basic support for this type of data refreshing still require a significant amount of manual effort. They neither track the generations that exist for an environment nor generate the restore jobs needed to refresh the sandbox. Just as noteworthy is the fact that it is not easy to manage the backups. A cloning tool that lacks an intuitive user interface to automate the work is labor intensive and error prone.

Another important feature is the handling of structures. If the structure of a table has been changed between different backups, the restore should not only restore the data but also

the corresponding structure. A state of the art cloning tool must offer this capability. Another important feature involves restoring an older backup into an environment that has been changed. The cloning tool should support restoring the data taken from an older structure into the new one.

8. Data Cloning Between Different Machines

For security reasons, many shops isolate the production environment from all other environments. This can make providing current test data from a production system quite difficult because it's not possible to copy the data directly to the testing environment. Usually the data is copied from production to a pool of shared volumes, and from there into the testing environment. Sometimes there is no shared pool, or it is not large enough for temporarily storing all the data. A cloning tool should offer multiple ways to move the data in order to deal with these possible restrictions.



One solution that is offered by BCV5 is to use TCP/IP to copy the data over a network connection. A small server that allows sending and receiving data over the network is started on the target machine. The copy facility is a network client that reads the data from the production environment and sends it to the target (testing, development) using the server component.

The remote copy technology offers the following advantages:

1. The data can be copied from a completely isolated environment (given that TCP/IP is available) into a test or development environment to use that data for testing purposes, or to recreate a production error.
2. Compared with a two-step copy (first from production to the shared pool, and then from the shared pool into the target), this method is faster and uses less disk space because it can copy directly from source to target, without storing any intermediate data.

Finally, copying over a network connection is completely transparent for the user. The only work required is specifying the source and target subsystem, choosing the processing options and selecting the objects. The tool does all the remaining work automatically. It recognizes that the data must be copied using TCP/IP and uses the server component to transfer the data from source to target. No manual intervention or special handling is required.

9. Other challenges that should be handled

Many shops have been using DB2 for a number of years. They might have started with DB2 version 4, and migrated to newer releases as these became available. Currently the latest version is DB2 version 10. Each release offered a set of new features, but from time to time IBM dropped support for certain older features. For instance, DB2 version 9 no longer supports creating a particular type of tablespace (simple tablespaces). A common situation finds that production is still using an old format,

but there is a need to copy that data into a test environment where it is not possible to create that kind of tablespace. The challenge is to convert the objects new format without impacting or touching production at all. A cloning tool should automatically handle such a situation and should convert such tablespaces on the fly without any user intervention. Of course, an UNLOAD/LOAD based process handles such a conversion, but at a steep cost in CPU consumption and runtime. Almost every site battles the problem that the nightly batch window is too small to copy the test data using UNLOAD/LOAD. Therefore, the cloning tool should provide a better way to do this.

As many shops refresh their test environments periodically, it should be easy to integrate the cloning process into a job scheduling system that executes the copy process every night or every weekend. This poses certain restrictions on the jobs. They should not use inline datasets, and they should not change between subsequent executions of the same copy process, even if the source or target environments change. In a nutshell, the JCL of the jobs should be static, which saves a lot of work when integrating the jobs into a scheduler.

10. Pricing

The cloning tool should be affordable. Even the best cloning tool must pass a strict cost-benefit analysis. BCV5 is priced extremely competitively. Both site licenses and licenses for separate machines are available. It does not require expensive hardware-assisted copy facilities, however, if they are available, BCV5 can use them. The product itself has a very favorable ROI. The flexible licensing options allows customers to only pay for the features they need or want.

Summary

It is no longer state of the art to refresh test data infrequently or in unison with “slow system demand times”, such as weekends or planned outages. With both human and machine resources being valuable commodities that should not be wasted, the demand for a cloning tool that handles all the challenges discussed is a must have item. The BCV product family is one of the market leaders for DB2 data cloning. From BCV4’s full DB2 subsystem clones to BCV5’s object level cloning and refresh, the BCVn solutions offer fast, fully automated processes that are both reliable and make good economic use of the available resources.

Unique capabilities, unparalleled support and a history of product enhancements make our products the premier choice in fast DB2 data cloning, copy and refresh.