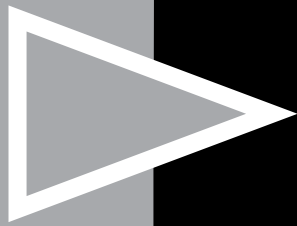




ACTIVATE BUSINESS WITH THE POWER OF I.T.™



Simplifying Database Administration with BMC CHANGE MANAGER for DB2

Table of Contents

The Administrative Challenge of Change.....	1
> BMC CHANGE MANAGER for DB2 — Compare Feature.....	1
Compare Input	1
Choosing Appropriate Input Sources.....	1
> DB2 Catalog.....	2
> DDL files	2
> Baselines	2
> Migrate worklists	2
Developing a Baseline Strategy	2
Compare Output Sources	3
> CDL Implementation	3
Overview of the Compare Process.....	3
Database Management Scenarios	4
> Scenario 1 — Migrate new set of DB2 structure definition changes.....	4
> BMC CHANGE MANAGER Solution.....	4
> Scenario 2 — Back out database definition changes.....	4
> BMC CHANGE MANAGER Solution.....	4
Summary	5

The Administrative Challenge of Change

Managing DB2 in today's business environment is more difficult than ever. Business dynamics have increased the rate of change and complexity within the typical IT environment. The explosive growth of data, the decline in experienced workforce, and the ability to respond to the needs of critical DB2 applications is a constant challenge. Objects are growing and changing in their number, size, and type. When a critical application requires a change, it affects the underlying objects. How can you ensure availability while maintaining data integrity?

Consider the following database management scenarios and how you might solve them manually:

- > Identify the structure differences between two databases and then make one of them look like the other. In other words, migrate a new set of DB2 structure definition changes from one subsystem to another.
- > Back out a series of database definition changes made to a database over the past few months, perhaps by different people, because the application developers need to back out their program changes.

You may know the exact manual process that needs to be performed to compare the "before" image of the object definitions that you want to change to an "after" image that already represents what you want the object to look like. You may have an electronic or hard copy of both images so that you can perform a manual comparison. However, because of the size of the objects, you might be trying to read and compare tens of thousands of lines of data definition language (DDL). You can do a manual comparison, but it will be time consuming, tedious, and error prone. And if you don't have a "before" or "after" image, then it will be a research project just to determine what the objects originally looked like, or should look like. BMC® CHANGE MANAGER for DB2 provides an automated, error-proof method.

BMC CHANGE MANAGER for DB2 — Compare feature

BMC CHANGE MANAGER for DB2 provides a Compare feature that performs complex tasks with accuracy, speed, and relative ease. Once you perform some initial planning and preparation, along with some understanding of BMC CHANGE MANAGER, you'll be able to develop a change methodology that can be implemented and used repeatedly with success. The following topics will familiarize you with BMC CHANGE MANAGER for DB2 components, terms, and concepts. This document will also cover how to perform the two scenarios previously discussed. The BMC CHANGE MANAGER for DB2 user guide and online can also help further tailor and fine tune a change methodology to suit

your needs. The first step is understanding the input sources that are valid for Compare.

Compare Input

Planning begins with knowing what the Compare feature of BMC CHANGE MANAGER for DB2 uses as input.

The Compare feature evaluates two sets of input to each other and differences are identified. The two Compare inputs can be from any of these sources:

- > DB2 catalog — a set of DB2 tables that are maintained by DB2, which contain definitions for all objects belonging to a subsystem
- > DDL file — a file containing DDL in a format that is executable in SPUFI. The DDL must consist of CREATE and ALTER statements. ALTER statements must reference objects that are previously named in CREATE statements (for example, ALTER statements that are used to create a primary key).
- > Baseline — a snapshot of the definition of objects captured at a particular point in time. A baseline is created in BMC CHANGE MANAGER and stored in the BMC CHANGE MANAGER DB2 tables. The objects in the baseline can reside in the DB2 catalog (a catalog baseline) or in a DDL file or worklist (a DDL baseline). In a catalog baseline, objects that are stored in the DB2 catalog are used as input to create the baseline. In a DDL baseline, object definitions that are stored in a DDL file or BMC CHANGE MANAGER migrate worklist are used as input to create the baseline.
- > BMC CHANGE MANAGER migrate worklist — DDL created by the Analysis component of BMC CHANGE MANAGER with control statements wrapped around the DDL. Like a DDL file, a migrate worklist consists of CREATE and ALTER statements. ALTER statements must reference objects that are previously named in CREATE statements (for example, ALTER statements that are used to create a primary key).

Choosing Appropriate Input Sources

The next planning step is to consider which Compare input sources best fit your environment and needs.

When you execute Compare, the input sources described previously are called Compare1 and Compare2. All possible combinations are valid, meaning that 16 different Compare types are valid. Because every combination works, you can choose the ones that work best for a given situation. Each of the 16 Compare types has advantages for different situations, so it is important to choose the right ones for your strategy or for the particular instance you have in mind.

The Compare input sources have the following strengths:

DB2 catalog

The catalog contains (unquestionably) the current definition of the objects in use. Catalog objects in different subsystems can be compared using DB2's Distributed Data Facility (DDF), which greatly expands the range of use for comparisons involving catalog objects.

A catalog-to-catalog comparison is the only Compare type that offers automatic change rules. Automatic change rules match the data base names, creators, and schema names in Compare1 to those in Compare2. This matching saves one step in the preparation process for performing a comparison, thus reducing the possibility of error and saving time.

DDL files

Software vendors often provide flat files containing DDL that is executable in SPUFI to create database objects for their applications. Compare can read DDL files as they are, without modification.

You can create a baseline of a DDL file. You can also do the reverse; in other words, you can convert a baseline to a DDL file by using BMC CHANGE MANAGER to create a baseline report.

Baselines

Baselines are the only Compare input source that enable Compare to recognize renamed objects. Every time you use a BMC CHANGE MANAGER Alter Work ID to rename an object or column, for instance, this event is recorded in the BMC CHANGE MANAGER tables. When a baseline is one of the inputs to Compare, Compare traces the history and recognizes the difference between a rename scenario and an add/delete scenario. Therefore, Compare knows whether the data in a column should be kept (a rename scenario) or discarded (an add/delete scenario).

Because a baseline can be taken of objects in the catalog, a DDL file, or a migrate worklist, the BMC CHANGE MANAGER baseline tables can be a central repository for all versions of your DDL, no matter how the DDL originated. Baselines in different subsystems can be compared using DDF, which greatly expands the range of use for comparisons involving baselines.

Migrate worklists

You can create BMC CHANGE MANAGER migrate worklists any time. It's a way of obtaining a copy of the DDL that was used to create database objects that currently exist in the DB2 catalog. You can cause a migrate worklist to be generated with different database names, creators, and schema names by using an outbound migrate profile with change rules at the time you create the worklist, thus avoiding the need to edit the DDL manually.

You can create a baseline of a migrate worklist.

Developing a Baseline Strategy

BMC CHANGE MANAGER baselines can act as a central repository for all versions of your databases and DDL. Because the DB2 catalog, DDL files, and BMC CHANGE MANAGER migrate worklists are all valid inputs to the baseline process, you have a lot of flexibility in creating a baseline. A baseline is a point to which your object definitions may be recovered. Generally speaking, you should create baselines of your database or DDL at meaningful sync points — at times when it is important for you to have a snapshot of what the database looked like at that moment.

A baseline can also be used as one of the inputs to a comparison that you may need at sometime in the future. When creating baselines, consider the project life cycle characteristics and the change management process in your environment. How are database changes migrated from one environment to the next?

- > If the changes to your database environment are based on a release theory, moving in organized groupings of changes in a first-in, first-out fashion, you may want to create a baseline after each set of database changes is applied
- > If database changes move from one environment to the next in no predictable order or grouping, you may not want to create a baseline every time a change is made. In this case, you may want to look at the broader picture and create a baseline only after major change implementations

When you create a baseline, include a comment that will help you remember the contents and significance of that particular baseline. Your comments, along with the baseline's name and the date and time that it was created, will help you in the future to select the particular baseline you need to use for a comparison.

Compare Output Sources

When you execute Compare, BMC CHANGE MANAGER produces three outputs:

- > A Change Definition Language™ (CDL®) file. CDL is a patented BMC proprietary language. The CDL file is a flat file, usually a PDS member that contains a record of the differences between the two sets of objects that were compared. In other words, it is a record of the structure changes that would have to be made to one set of objects to make them look like the other set. CDL is readable by the Import component of BMC CHANGE MANAGER. If a migrate or baseline profile is used to define the scope for the comparison, the scope rules appear as comments in the CDL file, serving as a reminder of the scope for the comparison. Multiple CDL files can be generated from one execution of Compare. This saves much manual effort when you need to send CDL files that are identical except for database name, creator, and schema name, to different environments (known as locations in BMC CHANGE MANAGER).
- > The Compare report shows you the differences between the Compare1 and Compare2 objects. Their original names and attribute values are shown side by side, so you can see their “before” and “after” images. Only one Compare report is produced, even if multiple CDL files are created.
- > The diagnostic log contains messages and return codes indicating the status of Compare processing.

CDL implementation

Once CDL has been created as the result of the Compare process, follow these steps to implement your database changes:

- > Use the Import component to read the CDL and store the changes in the BMC CHANGE MANAGER tables in an Alter Work ID. This achieves the same end result as if you had manually entered the changes using the Specification component. Importing CDL is an automated way to specify changes for DB2 objects. The Specification component can also be used to view the imported changes and make further modifications.
- > Use the Analysis component to create a worklist containing the DDL that is required to implement your changes.
- > Use the Execution component to execute the worklist.

Overview of the Compare Process

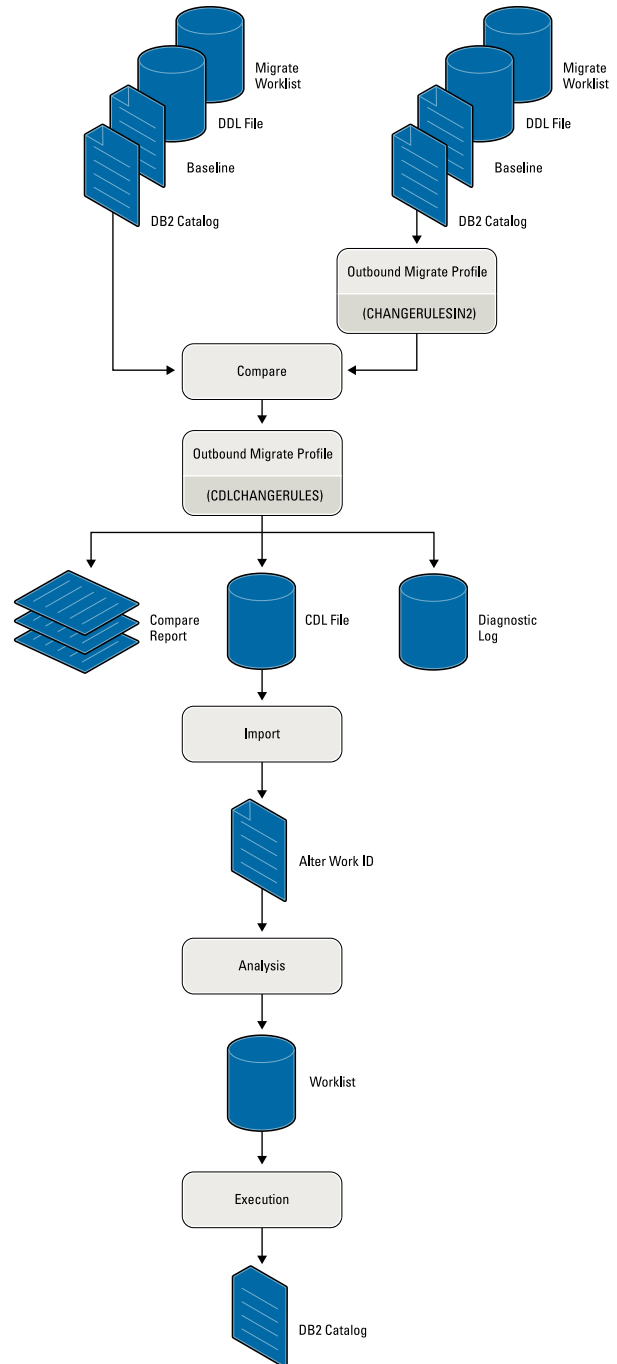


Figure 1. The flow of the Compare process in BMC CHANGE MANAGER for DB2

Database Management Scenarios

This section discusses how the Compare feature can solve the database management challenges. In each scenario, some Compare types require thought and advance preparation. For example, you will need to create baselines or save copies of DDL files or worklists at good syncpoints ahead of time, so that they are available to use them to perform a comparison.

Scenario 1 — Migrate new set of DB2 structure definition changes

Identify the structure differences between two databases, and then make one of them look like the other. In other words, migrate a new set of DB2 structure definition changes from one subsystem to another.

BMC CHANGE MANAGER solution

In this case, you are updating the current database to version that exists in the catalog. Because the database you want to change is usually defined in a DB2 catalog, one of the following Compare types would be the easiest way to accomplish the task. Consider the advantages of each type described earlier and what you have available for the second Compare input (Compare2).

- > Perform a catalog-to-catalog comparison if both databases (the “before” and “after” images) are already defined in the DB2 catalog. Because Compare uses DDF to retrieve objects from remote locations, a catalog-to-catalog comparison can be used even when the databases are defined in different subsystems.
- > Perform a catalog-to-baseline comparison if you create a baseline of your databases at crucial and meaningful times during a project’s life cycle. The comments you stored with the baseline when you created it will help you remember the significance each baseline.
- > Perform a catalog-to-DDL comparison when the Compare2 database is not defined in the DB2 catalog and you don’t have a baseline to use. You would need to have saved a copy of the DDL at the right point in time and remembered its significance to compare the catalog to the right version of the DDL. This comparison type is also useful if access to the catalog for Compare2 is restricted due to shop standards.
- > Perform a catalog-to-worklist comparison when the Compare2 database is not defined in the DB2 catalog and you don’t have a baseline to use. You would need to have saved a copy of a migrate worklist at the right point in time and remembered its significance to compare the catalog to the right worklist. This comparison type is also useful if access to the catalog for Compare2 is restricted due to shop standards.

Scenario 2 — Back out database definition changes

Back out a series of database definition changes made to a database over the past few months, perhaps by different people, because the application developers need to back out their program changes.

BMC CHANGE MANAGER solution

In this case, you are changing the database to a previous version of what is currently stored in the catalog. Because the database you want to change is defined in the DB2 catalog, a catalog-to-baseline comparison would probably serve you best in this case, provided you have a baseline created at the right point in time. Here’s why:

- > Baselines recognize renamed objects.
- > If you create full recovery baselines, you can restore both the structures and the data to a particular point in time.

You may decide that one of the following types of comparisons would serve you equally as well:

- > Catalog-to-catalog
- > Catalog-to-DDL file
- > Catalog-to-worklist

Summary

In today's complex DB2 for z/OS environments, it is imperative to have a well-thought-out change management philosophy and robust tool set. BMC CHANGE MANAGER for DB2 meets the challenges in the following ways:

- > You can maintain your change philosophy by allowing you to version your database environment at your choice of intervals, facilitating both upgrade to new versions and fallback to earlier versions.
- > You save significant time and effort by automating the process of migrating database changes to other environments while providing fallback capability. Besides saving time, automation helps to guarantee the accuracy of an otherwise manual, error-prone process.

With the right tools to implement change management procedures, you can efficiently manage the complexities of any DB2 for z/OS environment. This allows personnel to work together seamlessly, sharing administrative duties of managing complex environments and responding in the quickest way possible to implement change and maintain business performance objectives.



ACTIVATE BUSINESS WITH THE POWER OF IT.™

About BMC Software

BMC Software delivers the solutions IT needs to increase business value through better management of technology and IT processes. Our industry-leading Business Service Management solutions help you reduce cost, lower risk of business disruption, and benefit from an IT infrastructure built to support business growth and flexibility. Only BMC provides best practice IT processes, automated technology management, and award-winning BMC® Atrium™ technologies that offer a shared view into how IT services support business priorities. Known for enterprise solutions that span mainframe, distributed systems, and end-user devices, BMC also delivers solutions that address the unique challenges of the midsized business. Founded in 1980, BMC has offices worldwide and fiscal 2006 revenues of more than \$1.49 billion. Activate your business with the power of IT. www.bmc.com.

1-D87TBL

BMC Software, the BMC Software logos and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. All other registered trademarks or trademarks belong to their respective companies. ©2007 BMC Software, Inc. All rights reserved.



* 6 8 7 1 3 *