ALTER and CHANGE MANAGER®
for DB2®
Getting Started Guide

Supporting
Version 10.1 of Administrative Assistant for DB2
Version 10.1 of ALTER for DB2
Version 10.1 of CHANGE MANAGER for DB2
Version 10.1 of Database Administration for DB2

April 2011
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>713 918 8800 or</td>
<td>713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td>800 841 2031</td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) 713 918 8800</td>
<td>(01) 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2007-2011 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

DB2, DFSMS, DFSMSShm, IBM, MVS, OS/390, QMF, RACF, REXX, z/OS, and zSeries are the trademarks or registered trademarks of International Business Machines Corporation.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can

- read overviews about support services and programs that BMC offers
- find the most current information about BMC products
- search a database for issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
# Contents

## About this book
- Related publications .................................................. 16
- Conventions ................................................................. 16
- Syntax statements ...................................................... 18

## Chapter 1  Introducing ALTER and CHANGE MANAGER
- Introduction ........................................................................ 19
- Managing change ............................................................ 20
  - Identifying your environment ........................................ 20
  - Locating your environment ........................................... 20
  - Propagating your changes ............................................ 20
- Using ALTER and CHANGE MANAGER to manage change .......................... 22
  - Managing changes with ALTER .................................... 22
  - Managing changes with CHANGE MANAGER .................. 25
- Exploring ALTER ............................................................. 30
  - ALTER components ................................................... 32
  - ALTER objects .......................................................... 35
- Exploring CHANGE MANAGER .......................................... 37
  - CHANGE MANAGER components ................................. 40
  - CHANGE MANAGER objects ....................................... 43
- Solution integration ......................................................... 50
  - Administrative Assistant ........................................... 50
  - Database Administration ............................................ 50
- Where to go from here .................................................... 50

## Chapter 2  Using the ALTER and CHANGE MANAGER interface
- Overview ........................................................................... 51
- Using the ALTER or CHANGE MANAGER interface ......................... 52
  - Using the Main Menu .................................................. 53
  - Displaying the current environment .................................. 54
  - Using Fast Path Navigation .......................................... 54
  - Using ISPF commands ................................................ 55
  - Viewing BMC message information ................................. 57
- Achieving your goals with ALTER and CHANGE MANAGER ................... 57
  - Creating, altering, or dropping DB2 objects ....................... 57
  - Migrating DB2 objects ................................................ 80
- Where to go from here .................................................... 131
Overview .................................................................................. 134
Using options to control your environment .................................. 135
  Using the installation options .................................................. 135
  Using the user options .......................................................... 136
  Using the product options ...................................................... 137
  Using local keywords to override options ................................. 139
  Creating an additional installation options module .................. 139
  Putting it all together ........................................................... 140
Allocating data sets .................................................................. 141
Naming objects ....................................................................... 141
  Using wildcard patterns ........................................................ 142
  Using delimited identifiers ..................................................... 144
Using nonprintable or nonviewable characters ............................ 145
Controlling access to components ............................................. 145
Switching authorization ............................................................ 149
  Using the Execution security exit function codes .................... 149
  Testing the Execution security exit ......................................... 152
Restricting access to the worklist parallelism feature ................. 152
  Applying a zap ....................................................................... 153
  Specifying a general resource profile .................................... 153
Setting the MEMLIMIT system parameter .................................. 154
Modifying the ISPF skeletons and variables ............................... 154
Sizing the data sets .................................................................. 155
Maintaining the product tables .................................................. 156
  Maintaining the size of tables ................................................ 159
  Securing tables .................................................................... 160
Setting options ....................................................................... 163
  Setting the attributes for the panel display ............................. 164
  Setting the options for data set allocation ............................... 166
  Setting the processing options for Analysis ............................ 167
  Setting the processing options for Import .............................. 169
  Setting the processing options for Execution ......................... 171
  Setting the processing options for Compare ......................... 172
  Setting the processing options for Baseline ............................ 174
  Setting the options for the baseline report ......................... 175
  Setting the user options for CM/PILOT ................................. 176
  Setting the JCL options for job cards .................................... 180
  Setting the JCL options for STEPLIBs .................................... 182
  Setting the JCL options for static data sets ............................ 184
  Setting the JCL options for tapes ......................................... 188
  Setting the JCL options for temporary work data sets ........... 191
  Setting the JCL options for permanent data sets .................. 193
  Setting the JCL generation data group options ...................... 202
  Setting the JCL debugging, display, and Execution options ........ 203
  Setting the JCL utility installation options module name options . 207
  Setting the online reorg options .......................................... 208
  Setting the Execution worklist parallelism options ................. 209
  Setting the worklist options ................................................. 211
Setting user variables .................................................. 213
Creating a user POF ......................................................... 214
Updating a user POF ......................................................... 216
Using multiple POFs ......................................................... 217
Refreshing the initial POF ................................................. 218
Generating POF reports .................................................. 219
Reusing a POF in a subsequent installation ....................... 219
Overriding POF values in SLIBs ...................................... 221
Creating OUTPUT or TEMPLATE descriptors for copies in ALTER and
CHANGE MANAGER ..................................................... 223
Where to go from here ...................................................... 225

Glossary .................................................................... 227

Index ....................................................................... 249
Figures

Managing multiple environments .......................... 21
Maintaining and upgrading an application .................. 29
ALTER architecture ........................................... 31
CHANGE MANAGER components and objects ................. 38
CM/PILOT components and objects ........................ 39
Administrative products for DB2 panel .................. 52
CHANGE MANAGER Main Menu .......................... 53
BMC Environment panel ..................................... 54
Task List Menu .................................................. 59
Create, Alter, or Drop DB2 Objects panel ............... 59
Create WORKID panel ....................................... 60
Object Specification panel .................................. 61
Mixed List panel .................................................. 62
Table Detail panel .............................................. 62
Table Columns List panel .................................. 63
Mixed List panel showing changes ....................... 64
Analysis Alter Worklist Interface panel .................. 64
Analysis JCL Processing Interface panel .................. 65
Analysis input .................................................. 66
Analysis diagnostic output .................................. 67
CHANGE MANAGER alter-type worklist .................. 67
Execution JCL Build Interface panel ....................... 69
Execution Pre- and Post-Processing Interface panel ...... 70
Execution JCL Processing Interface panel .................. 71
Execution JCL for an alter-type worklist .................. 71
Execution results for an alter-type worklist .............. 75
Task List Menu .................................................. 82
Migrate DB2 Objects Using Specification panel .......... 82
Create WORKID panel ....................................... 83
Change Rules panel ........................................... 84
Edit Change Rules panel .................................... 85
Edit Rules Attribute Selection panel ....................... 86
Edit Change Rules panel—NAME attribute ............... 86
Change Rules panel—NAME attribute ....................... 87
Change Rules panel—OWNER attribute ..................... 88
Object Specification panel .................................. 88
Mixed List panel .................................................. 89
Mixed List panel with dependent objects ................. 90
Edit WORKID Migration Options panel .................... 91
Mixed List panel—database selected for migration .......... 92
Specifying the prefix for a dynamically allocated data set .......................... 224
Excerpt of batch Analysis JCL with POF keywords ................................. 224
Tables

Fast Path Navigation commands ................................................................. 55
ISPF commands ......................................................................................... 55
Guidelines for data sets ............................................................................. 141
Using wildcards .......................................................................................... 142
Wildcard characters for Specification filters ............................................. 142
Wildcard characters for scope rules ........................................................... 143
Wildcard characters for change rules ......................................................... 143
Examples of valid and invalid wildcard use for change rules .................... 143
Nonprintable or nonviewable characters .................................................... 145
ALTER and CHANGE MANAGER plans .................................................. 146
CHANGE MANAGER plans ......................................................................... 147
Execution plans .......................................................................................... 148
Execution security exit function codes ....................................................... 150
User-defined variables ............................................................................... 154
Data set sizing values and sources ............................................................. 155
Table prefixes ............................................................................................ 156
ALTER and CHANGE MANAGER tables .................................................. 157
SQL for updating rename table ................................................................... 161
Set up tasks ............................................................................................... 163
Data set sizing options .............................................................................. 185
Work data sets in the JCL cleanup job step ................................................ 187
Work data sets used by utilities ................................................................. 187
Values for 7-track tape drives .................................................................. 189
Prefixes for dynamically-allocated data sets ............................................. 195
Comment codes for data set sizing ............................................................ 205
Execution worklist parallelism options ....................................................... 210
Worklist sorting options ............................................................................ 212
About this book

The *ALTER and CHANGE MANAGER for DB2® Getting Started Guide* introduces you to the BMC Software ALTER and CHANGE MANAGER products for the DB2 DBMS.

To use this book, you should be familiar with the following items:

- IBM® DB2 Universal Database for z/OS® and OS/390®
- job control language (JCL)
- Interactive System Productivity Facility (ISPF)

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the Customer Support website at [http://www.bmc.com/support](http://www.bmc.com/support). Most product shipments also include the books on a documentation CD.

**NOTE**

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at [http://www.adobe.com](http://www.adobe.com).

The software also offers online Help. To access Help, press F1 within any product or click the Help button in graphical user interfaces (GUIs).
Related publications

The following related publications supplement this book and the online Help:

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation</td>
<td>• Administrative Products for DB2 Installation Guide</td>
<td>contain information about components of BMC products and solutions for DB2, and for installing the BMC solutions</td>
</tr>
<tr>
<td></td>
<td>• Administrative Assistant for DB2 Installation Guide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Database Administration for DB2 Installation Guide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• BMC Products and Solutions for DB2 for z/OS Installation Planning Guide</td>
<td></td>
</tr>
<tr>
<td>product use</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide</td>
<td>explains all of the components, objects, tasks, and functions of the ALTER and CHANGE MANAGER products</td>
</tr>
<tr>
<td></td>
<td>ALTER and CHANGE MANAGER for DB2 Reference Manual</td>
<td>explains the objects, components, commands, and keywords for the ALTER and CHANGE MANAGER products</td>
</tr>
<tr>
<td></td>
<td>Administrative Products for DB2 Messages Manual</td>
<td>contains descriptions and responses for the information, warning, and error messages that the Administrative products generate</td>
</tr>
<tr>
<td></td>
<td>DASD MANAGER PLUS for DB2 Reference Manual</td>
<td>contains information about the Space Estimation feature that the Specification component uses. These functions are available only if the BMC DASD MANAGER PLUS for DB2 product is installed.</td>
</tr>
<tr>
<td>notices</td>
<td>release notes</td>
<td>provides updates to the installation instructions and last-minute product information</td>
</tr>
<tr>
<td>online Help</td>
<td>ISPF Help panels</td>
<td>provides detailed information about options and terms</td>
</tr>
</tbody>
</table>

Conventions

This book uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.
• Variable text in path names, system messages, or syntax is displayed in italic text:

  `testsys/instance/fileName`

• The symbol `=>` connects items in a menu sequence. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.

• Change bars signify changes that clarify or correct existing information, or that provide new information corresponding to product changes. This book does not use change bars to denote editorial and formatting changes or typographical errors that have been fixed, unless these updates significantly affect your use of the information.

The following symbols represent components and data structures in processes:

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Symbol 1]</td>
<td>represent a collection of tables, such as the DB2 catalog, or internal storage. These symbols are also used to represent baselines.</td>
</tr>
<tr>
<td>![Symbol 2]</td>
<td>represents files that usually reside on DASD, although they might be allocated on a tape unit.</td>
</tr>
<tr>
<td>![Symbol 3]</td>
<td>represents unload data sets that usually are allocated on tape, although they might reside on DASD.</td>
</tr>
<tr>
<td>![Symbol 4]</td>
<td>represent ALTER or CHANGE MANAGER objects: work IDs, migrate profiles, and baseline profiles.</td>
</tr>
<tr>
<td>![Symbol 5]</td>
<td>represents components, the programs used to perform functions.</td>
</tr>
<tr>
<td>![Symbol 6]</td>
<td>represents the flow of information or data from one entity to another.</td>
</tr>
</tbody>
</table>
Syntax statements

The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with a name or value. If a variable is represented by two or more words, initial capitals distinguish the second and subsequent words.</td>
<td>alias&lt;br&gt;databaseDirectory&lt;br&gt;serverHostName</td>
</tr>
<tr>
<td>Brackets indicate a group of optional items. Do not type the brackets when you enter the option. A comma means that you can choose one or more of the listed options. You must use a comma to separate the options if you choose more than one option.</td>
<td>[tableName, columnName, field]&lt;br&gt;[-full, -incremental, -level] (Unix)</td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item.</td>
<td>{DBDName</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed items. In the example, you would choose either commit or cancel.</td>
<td>{commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items as many times as necessary.</td>
<td>columnName ...</td>
</tr>
</tbody>
</table>
Introduction

As DB2 applications become more structurally complex and more mission critical, the need to add and modify data structures, as well as the need for change management, increases significantly. In addition, the growing complexity of the DB2 environment has made the change process itself more difficult and costly. Without an effective change management tool, database administrators (DBAs) find that data structure changes and change management is complex, tedious, resource intensive, and error prone.
Managing change

Every organization, every environment, and every application is unique. Before you can use change management tools to help you manage changes to your data structures and your data, you might need to consider the following questions:

- What does my environment look like?
- Where are my environments located?
- How are my changes propagated?
- When are my changes propagated?

Identifying your environment

Most organizations use at least three environments for each of their applications: development, test, and production. What does your environment look like? Perhaps you maintain your application code and perform unit tests in a development system. Your test system might be used to perform system and stress tests, as well as simulate production. Your production system might be in single or multiple locations.

Locating your environment

Sometimes multiple environments are located on the same DB2 subsystem. The same DB2 subsystem can be used for multiple environments if you use different database names and you have different owners. Where are your environments located? Are they located on the same subsystem? Do they share DASD? Are they located in different cities or in different parts of the world?

Propagating your changes

In some organizations, multiple teams are responsible for developing or maintaining an application in a development environment. If the changes to the data structures are needed by each team, you might need to synchronize the data structures in the development environment before you migrate the entire structure to the test environment. How will you propagate these changes? Do you manage your change requests based on release cycles or by date?
One way to manage your development, test, and production environments is shown in Figure 1. As this diagram illustrates, if you need to create an environment, you can migrate your data structures and data. If the environment already exists, you can compare one environment to another to determine what changes need to be made to synchronize the data structures in both of the environments. Then, you can create, alter, or drop data structures in the environment, or simply maintain the environment.

For example, you could create your development environment based on your production environment or synchronize your development environment with your production environment. To create your development environment, you could migrate your production environment. To synchronize your development environment with your production environment, you could compare your development environment to the production environment. After your development environment is established, you could modify it or maintain it.
Using ALTER and CHANGE MANAGER to manage change

BMC provides two products to help you manage data structures and data for DB2 subsystems: the ALTER and CHANGE MANAGER products.

Managing changes with ALTER

ALTER provides a powerful solution to the problems of managing a DB2 environment. By automating and simplifying the change process, ALTER enables you to deal effectively with the demands of a constantly changing DB2 environment. ALTER uses an Interactive System Productivity Facility (ISPF) interface.

ALTER is designed for the single DB2 system that is beginning to encounter complex challenges. As the need for more change management functions develops, as with multiple DB2 subsystems, you can upgrade from ALTER to CHANGE MANAGER.

Overview

ALTER enables you to automate the following common tasks:

- create, modify, and drop data structures within a DB2 subsystem
- create new data structures by using existing data structures as templates
- determine the dependencies on changed objects and preserve those dependencies and their associated data
- migrate data structures and data within a subsystem or from one DB2 subsystem to another subsystem
- analyze the impact of changes by validating the changes against the DB2 catalog

If conflicts are detected, ALTER notifies you through error messages and warnings.

- implement a least-cost strategy for performing changes
  You can group your changes to minimize the work that is required to execute them.
- import data definition language (DDL) files and create the objects
- perform space estimation for table spaces and indexes
- provide passive management of authorizations, plans, and packages
- provide automatic data conversion for changes to column attributes
- handle requirements for data set allocations
- specify global structure changes when migrating objects

ALTER provides full management support for the following DB2 data structures:

- storage groups (stogroups)
- databases
- table spaces
- tables
- check constraints
- foreign keys
- triggers
- indexes
- unique constraints
- views
- synonyms
- aliases

When you use ALTER to specify changes for any of these data structures, the product automatically propagates the changes to any dependent objects. For example, if you change the name of a table, ALTER creates a corresponding change in the indexes, synonyms, and other dependent objects that reference the table under its former name.

ALTER uses BMC or IBM utilities in worklists when required. By using installed BMC utilities instead of IBM utilities, you can significantly enhance the performance of executing ALTER worklists. BMC utilities run faster, provide additional features, and might reduce the number of steps in a worklist.

You can use the following BMC utilities with ALTER:

- BMCSTATS utility of DASD MANAGER PLUS
- BMC COPY PLUS for DB2
- BMC CHECK PLUS for DB2
- BMC LOADPLUS for DB2
- BMC RECOVER PLUS for DB2
- BMC REORG PLUS for DB2
- BMC UNLOAD PLUS for DB2
The utilities must support the same version of DB2 as ALTER or CHANGE MANAGER. For more information, see the utility product’s reference manual.

**Changing data structures**

Within a DB2 subsystem, an application’s data structures need continual modification and extension because of changes in application requirements, performance tuning, and expanding uses of data. ALTER enables you to make changes to data structures in an easy, reliable, and automated way. You can efficiently perform changes while preserving data and, if necessary, convert the data to match the changes. ALTER also enables you to rebuild dependent objects and then propagate changes into those dependent objects. Without ALTER, these tasks can be extremely difficult to perform.

---

**WARNING**

BMC does not recommend using ALTER to modify the DB2 catalog, directory, and workfile objects.

---

**Migrating data structures**

If you have multiple DB2 subsystems, you might need to copy data structures from one subsystem to another. Most DB2 sites have separate subsystems for development, testing, and production. Many sites support multiple production systems, some of which are local and some remote.

ALTER offers the following functionality to simplify data structure migration:

- generates SQL CREATE statements from the sending subsystem for use on the receiving subsystem
- applies CREATE statements on the receiving subsystem
- moves data from the sending subsystem to the receiving subsystem

When you migrate an application’s data structures, ALTER assumes that these data structures do not yet exist on the receiving subsystem. ALTER cannot create new data structures if data structures with the same names already exist on the receiving subsystems.
Importing data structures

You use ALTER to process DDL files that are received from an external subsystem and apply the changes to your subsystem. The Import component converts these data structure definitions into change requests that are grouped together in a work ID. ALTER stores these change requests in the Change Definition (CD) tables.

Estimating space

ALTER provides space estimation capability that enables you to determine the amount of space that a table space or index will require based on the object definitions and their estimated usages. You can specify estimates of the number of rows in the table, and the average length of the rows. ALTER will use this information to project the number of tracks, cylinders, or blocks that are required to allocate the data set for that object.

NOTE

You cannot resize the DB2 catalog or directory tables.

Managing changes with CHANGE MANAGER

CHANGE MANAGER enables the DBA to effectively deal with the demands of a constantly changing environment involving multiple DB2 subsystems. Like ALTER, CHANGE MANAGER uses an ISPF interface.

Overview

CHANGE MANAGER includes all of the features in ALTER and provides the following additional capabilities:

- migrates data structure changes across multiple databases and subsystems
- determines changes to data structures and migrates those changes to one or more copies of the data structures
- captures and records structure definitions and data within a DB2 subsystem at a point in time (to establish a full-recovery baseline)
- recovers structures and data to a point in time defined by a full-recovery baseline within a DB2 subsystem
Managing changes with CHANGE MANAGER

- compares two versions of structure definitions to
  - determine the changes necessary to upgrade one version to another
  - selectively apply changes to copies of the data structures while preserving the uniqueness of each copy

- uses data modeling tool outputs to determine the changes to existing DB2 application structures

- reduces the volume of information that is needed to communicate changes by using a BMC language called Change Definition Language (CDL) to transmit the change information

- feeds changes that are made on a remote system back to the development system

- defines and stores reusable rules for making changes

- uses SQL-like statements to update, delete, and migrate data structures

CHANGE MANAGER provides full management support for the same DB2 data structures as ALTER does.

**Migrating only data structure changes**

*Change migration*, which means to migrate only data structure changes (instead of entire structures) to another copy of the structure on the same DB2 subsystem or on a different DB2 subsystem, enables you to update structures that have already been migrated to another copy. However, you must also be able to retain any structure modifications that were made locally after the structures were migrated. DB2 users often make local modifications—or variations from the control definition of the application—to meet the following needs:

- **Performance and usage tuning**

  Local tuning is a complex process that is based on the specific system that is involved and the performance and use of the application in that environment. Because separate sites often have different amounts of data and different transaction loads, as well as different CPU and DB2 environments, local tuning of application structures is common.

- **Security**

  Authorization requirements are associated with object ownership and, in some cases, with object names. As a result, the authorizations for one DB2 subsystem can vary significantly from the authorizations of another subsystem running the same application. In this situation, each subsystem requires local modifications to the application.
- Additional uses of data

Often, production systems in a DB2 environment need to use data that is beyond the fixed scope of an application. For example, the data in a payroll application might be needed for reports that the finance department develops. DB2 enables users to locally generate additional data structures (such as indexes, synonyms, views, aliases, and authorizations) to meet these needs.

Because local modification of applications is so common in DB2, users need an application management strategy that enables them to manage basic elements of the application globally, without compromising the elements that vary locally.

CHANGE MANAGER satisfies the requirement of migrating global changes while retaining local modifications by

- determining the changes made to the control version of the application
- applying only those changes to other versions while preserving all local modifications

CHANGE MANAGER can also determine the accuracy of the proposed changes before applying them. Because changes are allowed at the lowest level of an object’s definition, CHANGE MANAGER preserves and converts data as necessary to accommodate the requested changes.

Additionally, CHANGE MANAGER propagates changes to all dependent structures. For example, when you change a column name, CHANGE MANAGER propagates the change to any index and foreign key definitions that use the column, even though the index and foreign key definitions might be unknown to the sending DB2 subsystem.

**WARNING**

BMC does not recommend using CHANGE MANAGER or the CM/PILOT component of CHANGE MANAGER to modify the DB2 catalog, directory, and workfile objects.

### Recovering data structures

Changes to data structures might not produce the effect that you intended. For example, a change might significantly increase an application’s response time. In this case, you might need to restore the previous data structures. While DB2 has features for logging, backup, and recovery of data, it has no similar features for data structures. If you make an unwanted structure change, or if your changes fail or are unusable, you must fall back to the previous definition. To do so, you must have saved the structure definitions and data of the previous version.
CHANGE MANAGER enables you to capture a set of data structures from the DB2 catalog or DDL file and store the set in a baseline. A baseline can contain only data structures or data structures with their associated data.

Establishing baselines and recovering to them is called data structure recovery. You can reload the data that is stored in a full-recovery baseline after the structures have been recovered. You can also reload the data that exists in those structures at the time of their recovery. In addition, CHANGE MANAGER can convert current data to match the restored data structures.

**Recording and controlling changes**

Managing change also requires that you know what changes have been made to an application’s data structures during different periods of time. CHANGE MANAGER provides a comparison feature that can generate a file that shows the differences between two sets of data structures. Because the file is stored in a sequential data set or partitioned data set (PDS) member, you can view, edit, print, or store it.

**Feeding back changes**

Data structure changes might not flow in order from the development system, through testing, and into production. Because changes to the basic application definition can occur at any point in the cycle, you must be able to transmit changes back to the development system or to the control node. Transmitting changes from a remote system to the development system (or to the control node) is called change feedback.

To feed back changes, CHANGE MANAGER enables you to

- identify the changes made
- apply the changes at the development or control system

You do not need to feed back all of the changes to the development system. You should feed back only those changes that are part of the globally managed basic definition of the application.

**Using data modeling tools**

In many installations, the data structure design of the application is not controlled in a DB2 subsystem but in a data modeling tool (or Computer-Aided Software Engineering, or CASE, tool), data dictionary system, or other structure design repository. In this guide, the term data modeling tool refers generically to any external design tool.
Most data modeling tools generate an application’s DDL. Changing an application’s data structures generates new DDL, which you can use to create the data structures in the DB2 catalog. Although data modeling tools can improve the productivity of DB2 developers, the use of these tools creates the following problems:

- If you make an emergency change in a production system, including that change in the data modeling design is difficult and usually requires a manual entry. You can use host-based DB2 tools to make emergency changes (or to define local structure elements not supported by a data modeling tool), but the changed structures will no longer be synchronized with the data modeling design.

- Often, communication between a data modeling tool and DB2 is one way. You can design structures on the data modeling tool and transmit the structures to DB2, but updating the data modeling design after making changes at the DB2 node is difficult.

- Most data modeling tools cannot selectively modify data structures. Because these tools can transmit only a complete application definition, they might be useful only for the initial design of an application. Maintenance and upgrades are then performed at the DB2 level, as illustrated in Figure 2.

Figure 2  Maintaining and upgrading an application

- Due to their ease of use and built-in design intelligence, data modeling tools are popular for managing global application data structure definitions. However, the external DB2 specifications that a data modeling tool produces might be incomplete or might fail to reflect local elements of the design.
You can address these problems by using CHANGE MANAGER to migrate changes between the data modeling definition of an application and the DB2 definition. CHANGE MANAGER enables you to

- establish control points for the application’s data structures and determine the changes that are made between those control points
- implement design changes in DB2 without losing either the application’s data or its local data structure modifications
- transmit changes from DB2 back to the data modeling tool after making global changes to the DB2 version of the application’s data structures

**Exploring ALTER**

ALTER provides management of data structures and data for a DB2 subsystem. The product consists of several components that allow you to create, modify, and drop DB2 data structures while preserving data and dependent objects.

Figure 3 illustrates the architecture of the ALTER product. The components and the objects of ALTER are discussed in the sections that follow.
Figure 3  ALTER architecture

- Front End
  - Work ID
    - Specification
      - CD Tables
        - DB2 Catalog
          - Analysis
            - Worklist
              - JCL Generation
                - Execution JCL
                  - Execution
                    - CM Tables
                      - DDL
                        - Import
Most of the components that ALTER uses to perform its tasks can run in the foreground or as batch jobs. ALTER consists of the following components:

- Front End
- Specification
- Import (DDL only)
- Analysis
- JCL Generation
- Execution

Front End

The Front End component is a dialog panel that you use to access and control ALTER. From the Front End component, you can create, analyze, and execute work IDs, and build JCL in foreground or batch mode.

Specification

The Specification component enables you to create or edit data structure change or migration requests. The component stores the change or migration requests in a work ID. With Specification, you can perform the following tasks:

- create data structures
- change data structures
- delete data structures
- migrate data structures
- estimate the space required for a table space or index

The Specification component lists objects from the DB2 catalog (or from a copy of that catalog if your site uses catalog indirection). You can change these objects, delete them, or create new objects either from the list of objects or from the Command line of some Specification panels. Changes are not made to the data structures until you execute the change and migration requests.

**NOTE**

Although not depicted in Figure 3, the Front End component interacts with other ALTER objects and components. For more information, see “Front End” on page 32.
Import

You can import data structure definitions that are stored in a DDL file. The Import component converts these data structures into change requests in a work ID. The Import component provides a batch function similar to that provided by the Specification component with limited validation checking.

Analysis

After you create change or migration requests in a work ID, the Analysis component checks the requests for validity with the DB2 catalog, develops an optimal implementation strategy, and generates a worklist. The worklist contains the DB2 utility commands, AMS commands, security commands, and SQL statements that are necessary for implementing the requests. In addition, the Analysis component propagates changes to dependent structures. The Analysis component can be run in the foreground or in batch using the JCL Generation component.

The Analysis component performs the following validity and compliance checks:

- verifies that the data structures of changing objects have not changed in the DB2 catalog since the present changes were requested
- checks the change requests against other requested changes and warns of any conflicts
- checks the change requests for compliance with DB2 rules and reports conflicts
- develops the most efficient strategy to implement the requested changes or migrations
- locates all dependent structures
- binds or rebinds plans and packages
- migrates plan and package authorizations
- checks for objects that are indirectly affected by requested changes

**NOTE**

The Analysis component warns you if a drop or alter change request will result in the loss of objects, and also warns you of views and plans that might require modifications.
If no errors are found, the Analysis component generates a worklist, stores the worklist or associated data set name with the work ID, and uses it as the default worklist when you run the Execution component. The Analysis component also creates diagnostic output in ALUPRINT and provides authorization switching.

**JCL Generation**

The JCL Generation component works with the Front End component to construct a job control language (JCL) file for running the Analysis and Import components in batch. Using symbolic variables, the Front End component resolves all data set names that appear on the interface panels of the components. These names include the names of input files, JCL files, and the diagnostic output files.

**Execution**

The Execution component performs the commands that are contained in the worklist that the Analysis component generates for change or migrate requests. Although the tasks that are required to implement a worklist vary depending on the complexity of the actions involved, the Execution component performs the following basic functions:

- unloads DB2 data
- runs IBM or BMC utilities
- executes DB2 commands and Access Method Services (AMS) statements
- establishes synchronization (sync) points to accommodate restart, if needed
- creates diagnostic output in AEXPRINT
- performs audit and recovery logging, if requested
- performs authorization switching to ensure that DB2 objects are created with the correct owner and creator

The Execution component is the only component that makes physical changes to DB2 data structures.
ALTER objects

ALTER uses the following objects to perform its tasks. This section describes each of the objects in detail.

- work IDs
- worklists
- unload data sets
- DDL files
- internal tables

**Work IDs**

Work IDs enable you to identify, control, and track a unit of work. Work IDs contain change or migration requests that are stored in the CD tables. Creating or selecting a work ID is usually the first step in performing any type of change or migrate process.

ALTER uses the following types of work IDs:

- alter-type, which performs the change process
- migrate-type, which performs the migrate process
- receive-type, which receives a worklist that is generated by a migrate-type work ID that contains the migrated data structures and data from a different subsystem

You can request any number of changes or migrations with a single work ID. You can enter these requests all in one session or across many sessions. You can also delete or modify requests if you make a mistake or change your mind. Any number of users can request changes or migrations with the same work ID.

**Alter-type**

You use an alter-type work ID to change or modify data structures, drop objects, or create new data structures.

**Migrate-type**

You use migrate-type work IDs to specify the data structure definitions that are to be migrated from one DB2 subsystem to another, or to duplicate structures within the same subsystem. Migrate-type work IDs do not modify existing data structures.

Migrate-type work IDs can contain change rules to modify database object attributes and migrate options to specify the default dependencies to include in a migration. The change rules compare the object type and attribute to those in the migrated data structures and apply the rule when a match is found.
The migrate process provides a way to create new data structures from data structures that already exist on a subsystem. You can create the new data structures on the same subsystem if the objects are renamed. You can rename the objects through the Specification component or with change rules in the work ID.

**Receive-type**

You use receive-type Work IDs to implement migrated data structures and data that are being created on a different subsystem.

**Worklists**

The worklist data set contains all of the commands and SQL statements that are needed to execute the changes or migrations specified in a work ID. The Analysis component generates the worklist commands in a specific order. You can edit the worklist, but changing the order of the commands might yield incorrect results.

The worklist can contain any of the following items:

- SQL statements that perform the change or migration requests, including the dropping and re-creating of all affected objects and the restoring of dependent objects that are lost as a result of a change or that propagate the changes into other objects

- DB2 and utility commands that implement requested changes and migrations, including
  - unloading data
  - reorganizing table spaces and indexes
  - reloading data
  - collecting statistics
  - making image copies
  - rebuilding indexes
  - checking the data
  - rebinding application plans and packages (if an alter-type work ID)
  - binding plan and package authorizations (if a migrate-type work ID)
  - performing other authorizations

- AMS commands for data set allocation and deletion that support VCAT-defined partitions
Unload data sets

ALTER uses unload data sets to store data while dropping and rebuilding DB2 objects.

DDL files

ALTER does not generate DDL files. ALTER can use DDL files to import data structures.

Internal tables

ALTER uses the following types of internal tables to perform its tasks:

- CD tables, which store change and migration specifications for each work ID
- CM tables, which store work ID, change rule, and sync data

Exploring CHANGE MANAGER

CHANGE MANAGER consists of all of the ALTER components, as well as additional components that allow you to perform tasks across multiple subsystems.

Figure 4 illustrates the architecture of the CHANGE MANAGER product. The components and the objects of CHANGE MANAGER are discussed in the sections that follow.
The shaded icons represent the components and objects that are unique to CHANGE MANAGER. The CM/PILOT component of CHANGE MANAGER and the interaction of the Front End component with other CHANGE MANAGER objects and components is not depicted. For an illustration of the CM/PILOT component, see Figure 5. For more information about the interaction of the components, see page 32.
Figure 5  CM/PILOT components and objects

- Task ID
- Script
- Work ID
- Application
- DML
- CP Tables
- CM/PILOT Worklist
- JCL
- CHANGE MANAGER Worklist
CHANGE MANAGER components

Most components that CHANGE MANAGER uses to perform its tasks can run in the foreground or as batch jobs. CHANGE MANAGER uses the following components:

- Front End
- Specification
- Baseline
- Compare
- Import
- Analysis
- JCL Generation
- Execution
- CM/PILOT

The Specification and Analysis components provide the same functionality as they do in ALTER. For a description of these components, see “ALTER components” on page 32.

Front End

In CHANGE MANAGER, the Front End component can also maintain migrate and baseline profiles, and interact with the Baseline, Compare, and Import components.

Baseline

The Baseline component captures a set of DB2 structure definitions from either the DB2 catalog, a DDL file, or a migrate worklist at a specific point in time. The captured set of data structure definitions is called a baseline. If the structures are defined in the DB2 catalog, a baseline can also capture the data and authorizations that are associated with those structures. A baseline of both data structures and data is called a full-recovery baseline.
Baselines act as control points during data structure management. Baselines establish a static set of data structures for an application version. If you make a change with unwanted results, you can restore the data structures. If you establish a full-recovery baseline, you can also restore the data. You can also restore the data structure back to a prior baseline and convert the current data to those structures.

You can also use baselines for version comparison. For example, when you first install an application, create a baseline. At a later time, perform a comparison between the baseline and the DB2 catalog. This comparison creates a CDL file that shows any modifications that have been made to the application’s data structures since the initial installation.

**NOTE**
A baseline is also a CHANGE MANAGER object.

### Compare

The Compare component determines the differences between two sets of data structures and then generates a CDL file. The generated CDL file contains all of the changes that the comparison found between the two sets of data structures. You can compare data structures that are stored in a DDL file, work ID, baseline, worklist, or DB2 catalog. The inputs to the Compare component are called the *primary* and *secondary input sources*. The CDL file contains the changes that, if applied to the primary input source, would produce the data structures of the secondary input source.

You can use the generated CDL file in any of the following ways:

- process the file as a set of change requests for the current subsystem
- save the file as a record of the changes made
- import the file to a different subsystem in order to update a separate version of the data structures

CDL has advantages over DDL in automating the process of updating data structures. CDL allows more types of modifications to data structures and, unlike DDL, can retain local modifications to those structures.

### Import

You can import data structures that are stored in a CDL file or DDL file. The Import component converts the data structures in the files into change requests in a work ID. CHANGE MANAGER stores these change requests in the CD tables. The Import component provides a batch function similar to that provided by the Specification component with limited validation checking.
JCL Generation

In CHANGE MANAGER, the JCL Generation component also generates JCL for the Compare, Baseline, Baseline Report, and CM/PILOT components.

Execution

When CHANGE MANAGER is used in the BMC Database Administration for DB2 solution, the Execution component can enable a worklist to execute in parallel. That is, the component executes portions of the worklist concurrently. For information about the Database Administration solution, see "Database Administration" on page 50.

CM/PILOT

The CM/PILOT component automates DB2 change management tasks. With CM/PILOT, you do not need to decide which CHANGE MANAGER processes are required for a task or the sequence in which you need to complete them—CM/PILOT provides scripts to guide you through the process.

The CM/PILOT component enables you to choose from several predefined scripts or to create your own scripts. You can copy scripts, including those that CM/PILOT provides, and modify them to meet your needs. You can also edit, browse, delete, and perform other maintenance tasks on scripts.

CM/PILOT enables you to quickly and easily use the vast change management power of CHANGE MANAGER. By following the dialog panels that are provided in the CM/PILOT scripts or in the scripts that you create, you can specify, analyze, and execute an Analysis worklist to perform the following tasks:

- change data structures
- migrate data structures
- migrate only data
- receive data structure changes
- receive DDL to create data structures
- create full-recovery baselines
- recover data structures with current or old data
- select a user-defined script
- replicate work IDs

When you want to perform a change management task, CM/PILOT determines the components that CHANGE MANAGER will use and ensures that they are run in the correct sequence. CM/PILOT enables you to create tasks that can be done later by someone else or through job scheduling. By reusing these tasks, you can ensure that the change management task is done the same way every time.
The CM/PILOT component enables you to easily prepare for various CHANGE MANAGER tasks that can be processed on demand. CM/PILOT dialog panels are task oriented and easy to use. The panels guide you through the CHANGE MANAGER components that are necessary to accomplish the specific tasks and prompt you for the information that is needed to build worklists that can be processed at any time.

CM/PILOT simplifies the decision-making process and enables you to prepare change management tasks that are performed repeatedly. It also lets you create change management tasks that can be performed by less-experienced DBAs. And with its unique script-locking feature, CM/PILOT lets you create script steps that cannot be modified and can only be run as you have specified.

You can also use two CM/PILOT scripts to help you create SQL-like Data Manipulation Language (DML) statements to update, delete, and migrate data structures. You can also import DML statements into a work ID. After the statements are imported, CHANGE MANAGER creates entries in the CD tables as if they had been requested in the Specification component.

### CHANGE MANAGER objects

CHANGE MANAGER uses the following objects to perform its tasks. This section describes each of the objects in detail.

- work IDs
- worklists
- unload data sets
- baselines
- profiles
- CDL files
- DDL files
- internal tables
- task IDs
- scripts
- applications
- DML
- CM/PILOT worklists

The work ID and unload data set objects are the same objects as in ALTER. For a description of these objects, see "ALTER objects" on page 35.
**Worklists**

When CHANGE MANAGER is used in the Database Administration solution, the worklist can also contain keywords that control the sequencing of parallel processes. For information about the Database Administration solution, see “Database Administration” on page 50.

**Baselines**

A baseline contains a set of data structures and their associated authorizations, taken at a specific point in time, from the DB2 catalog, a DDL file, or a worklist. CHANGE MANAGER uses structure-only and full-recovery baselines:

- A structure-only baseline contains only data structures.
- A full-recovery baseline contains data structures and data.

The types of input to a baseline are as follows:

- DB2 catalog
  
  A catalog baseline takes its input from a subset of the DB2 catalog. See “Baseline profiles” on page 45.

- DDL file
  
  A DDL baseline takes its input from a file containing SQL DDL statements.

- Worklist file
  
  A worklist baseline takes its input from a worklist that CHANGE MANAGER or another BMC product produces. A worklist baseline is identical to a DDL baseline except for the input source file type. See “Baseline profiles” on page 45.

**Profiles**

Two types of profiles are used in CHANGE MANAGER: baseline profiles and migrate profiles. Profiles are a collection of scope rules, change rules, and locations that enable you to define and control the Analysis, Compare, and Baseline processes.

You use profiles to

- select a set of objects
- customize changes to objects that are migrated to other subsystems
- create baselines of application data structures
Profiles enable you to automate changes instead of selecting objects individually for migration or making individual changes to object structure definitions. You can create, copy, edit, and reuse profiles.

**Baseline profiles**

You use baseline profiles to create, name, and maintain baselines of application data structures. You can automatically manage the baselines that you want to keep and those that you want to delete by specifying delete and retain values in the baseline profile. The two types of baseline profiles are catalog and DDL (or worklist) profiles.

- **Catalog baseline profiles**

  The profile for a catalog baseline must contain scope rules that select the DB2 objects to include in the baseline or comparison. The baseline profile can explicitly specify the scope rules or can reference the scope rules in an outbound migrate profile.

  Scope rules are used only when a process is being performed on the DB2 catalog. The scope rules serve to create a subset of the DB2 catalog.

- **DDL or worklist baseline profiles**

  When defining a DDL baseline profile, you define only header information. The scope is implicitly defined by the set of objects in the DDL or worklist.

**Migrate profiles**

You use migrate profiles to select a set of objects and to customize changes to objects that are migrated to different locations. The two kinds of migrate profiles are inbound and outbound.

- **Inbound Migrate Profiles**

  The Import component uses inbound migrate profiles to make input from another system compatible with the receiving subsystem’s version of the application. Inbound migrate profiles contain only change rules that modify imported data structure definitions.

  The change rules are applied as the CDL file or DDL file is imported. For example, if a change rule specifies changing the database name DEMO* to ACM*, the DDL statement CREATE DATABASE DEMOHRS is read and the CD table entries are generated with a CREATE DATABASE ACMHRS. The profile’s change rules are applied to the imported data structure definitions before CD table entries are created.

  You cannot specify scope rules or locations with an inbound migrate profile. The file being imported identifies the scope.
Outbound Migrate Profiles

You use outbound migrate profiles to generate worklists or CDL files. The outbound migrate profile can contain the following items:

— a scope for specifying the DB2 objects to include in the operation
— locations for defining multiple clones of an application
— change rules for defining attribute changes to be automatically applied to objects in each clone

You can group locations if they reside in the same DB2 subsystem.

When an outbound migrate profile is used as input to Analysis, a worklist is generated for each group specified. One CDL output file is generated for each group when an outbound migrate profile that specifies locations is used as an input to the comparison.

The Compare process can also use the scope rules of an outbound migrate profile to select a subset of objects from the DB2 catalog to use in a comparison with a DDL file or worklist.

Scope rules

When you define scope rules for a profile, you are specifying which objects to fetch from the catalog. You can reference scope rules when you perform a comparison, create a baseline, or analyze a work ID.

In a catalog baseline profile, scope rules specify the DB2 objects to include in the baseline. The scope rules of an outbound migrate profile can specify the objects to migrate or the DB2 objects to include in a comparison. Scope rules can include DB2 objects in the final set of objects to be processed or can exclude them from the final set.

Scope rules can select a specific object or select groups of objects (through the use of wildcard characters) as the set or sets of objects on which to operate. Scope rules can also determine which dependent objects to include and to exclude. For example, a scope rule could include database ABCDEF and its dependent table spaces, tables, indexes, foreign keys, and views, but exclude its dependent synonyms, aliases, or data.

You can define scope rules for both migrate profiles and catalog baseline profiles. You can follow the same procedure to create a set of scope rules for each type of profile. You can also use the same set of scope rules for different profiles. For example, when creating a catalog baseline profile, you can reference the scope rules of a migrate profile. Likewise, a migrate profile can reference the scope rules of a catalog baseline profile. This feature enables you to use the same list to select the same objects for either migration or a baseline.
The Compare component uses the scope rules of a catalog baseline profile when it compares a catalog baseline to the DB2 catalog. The Compare component finds the baseline profile by using the established baseline name, and then uses the scope rules of the catalog baseline profile to select the objects to compare. Compare can also use the scope rules of a specified profile when it compares a DB2 catalog to a DDL file or to a worklist.

### Change rules

Change rules modify the attributes of existing DB2 object structure definitions. The change rules compare the object type and attributes to those in the migrated data structures. The change rules are applied when the attributes and the names match. For ALTER, you can define change rules within a migrate-type work ID. For CHANGE MANAGER, you can define change rules within a migrate-type work ID or a migrate profile.

When you define change rules within a migrate profile, the rules are processed for each object that is defined in the scope of the profile. You can streamline and automate changes to sets of object structures by defining change rules for migrate profiles. Because migrate profiles are used in different types of migration processes, special consideration of how change rules work can significantly contribute to the efficient performance of these processes.

Change rules allow you to easily specify and repeat common changes. For example, in many DB2 installations, the owner of a table changes when the table is moved from a test system into production. A migrate profile can perform the changes that are necessary to migrate the structures from the test system into production. This profile can include a change rule that changes all of the tables with owners that match TEST to PROD.

Change rules can perform the following tasks:

- change the names of objects
- change the attributes of a specified object or group of objects
- include specific volumes in a storage group, or exclude volumes from a storage group
- exclude one or more columns from a table
- specify that table spaces and indexes that have a STOGROUP attribute be defined to use VCAT on the receiving database
- suppress changes to altered objects (Compare only)
Locations for outbound migrate profiles

Locations, which are defined only in outbound migrate profiles, enable you to create or maintain multiple clones of an application schema. When an outbound migrate profile contains locations, a worklist or CDL is generated for each location or DB2 subsystem that is specified.

For example, if your installation supports systems in New York, Dallas, and Atlanta, you can create a profile with a location for each system. Each location can have change rules that tailor object names and attributes for that system. Each location can have its own change rules, or can reference the change rules of another location within the profile.

You can also group several locations under a single Group ID to generate just one worklist or one CDL file. Using the Group ID enables you to execute all of the work for a single subsystem in one process. This action is useful when you are migrating or updating several clones of a structure that reside on the same subsystem.

For example, if a development facility has 25 developers, each of whom have their own copy of the application schema, you could use a Group ID to migrate all 25 copies of the schema within a single worklist.

CDL files

CDL is a BMC proprietary language that you use to specify changes to DB2 data structures. You can use CDL files to transmit data structure changes between subsystems or to provide a record of changes to data structures.

CDL files specify the changes that are required to transform a primary set of data structures to a secondary set of data structures. You use CDL files in the change, import, and recovery processes.

DDL files

CHANGE MANAGER can generate executable SQL DDL statements (in SPUFI format) when generating a baseline report. DDL files are used as input by the Import, Baseline, and Compare components.

Internal tables

CHANGE MANAGER uses the following types of internal tables to perform its tasks:

- BL tables, which store information about baselines (data for full-recovery baselines is stored in external data sets)
- CD tables, which store change and migration specifications for each work ID
CM tables, which store work ID, change rule, and sync data

CP tables, which store information about CM/PILOT task IDs, applications, scripts, and DML

Access to most of the internal tables is controlled by the granting of EXECUTE authority on the appropriate plans.

Task IDs

A task ID is a unit of work in the CM/PILOT component. Each task ID has a unique name and contains information that you provide through the panel to perform a CHANGE MANAGER process. The information that you provide is based on the script you select for the task ID. All work in CM/PILOT is accomplished by processing task IDs.

Scripts

The panels that the CM/PILOT component displays are based on predefined steps called scripts. Each script contains ordered steps that prompt you for the information that is required to perform a change management task. CHANGE MANAGER provides you with several scripts; you can also create your own script, or copy, change, and rename any of the scripts that are supplied by CHANGE MANAGER.

Applications

In the CM/PILOT component, you can associate a group of CHANGE MANAGER profiles that are used repeatedly for the change management tasks of a specific DB2 application. This association is called an application.

DML

In the CM/PILOT component, you can use DML statements to update, delete, and migrate data structures. You can change or migrate multiple objects in a single DML statement. The panels for two of the CM/PILOT scripts help you create the DML statements.
CM/PILOT worklists

A CM/PILOT worklist is a data set that contains the ordered commands, keywords, and parameters that are needed to process a task ID. The format of the CM/PILOT worklist is similar to an Analysis worklist. The JCL that processes the CM/PILOT worklist contains the names of the output data sets that an Analysis worklist uses and the associated Execution JCL. When a CM/PILOT worklist is successfully processed, the result is the creation of an Analysis worklist and the Execution JCL to process the worklist.

Solution integration

ALTER and CHANGE MANAGER are also components of BMC solutions. ALTER is a component of the Administrative Assistant solution. CHANGE MANAGER is a component of the Database Administration solution.

Administrative Assistant

The Administrative Assistant solution enables users of all experience levels to navigate through the DB2 catalog quickly and to manage a complex DB2 environment easily. For more information, see the Administrative Assistant for DB2 Release Notes and the Administrative Assistant for DB2 Installation Guide.

Database Administration

You can use the Database Administration solution to manage your DB2 databases quickly, efficiently, and effectively. For more information, see the Database Administration for DB2 Release Notes and the Database Administration for DB2 Installation Guide.

Where to go from here

For an overview of the interface for ALTER and CHANGE MANAGER and an application of the features of the products, see Chapter 2, “Using the ALTER and CHANGE MANAGER interface.”
Using the ALTER and CHANGE MANAGER interface

This chapter presents the following topics:

Overview ............................................................ 51
Using the ALTER or CHANGE MANAGER interface .................. 52
  Using the Main Menu ........................................ 53
  Displaying the current environment .......................... 54
  Using Fast Path Navigation ................................. 54
  Using ISPF commands ..................................... 55
  Viewing BMC message information .......................... 57
Achieving your goals with ALTER and CHANGE MANAGER ............ 57
  Creating, altering, or dropping DB2 objects ................ 57
  Migrating DB2 objects ................................... 80
Where to go from here ........................................ 131

Overview

ALTER and CHANGE MANAGER are designed to help you and your organization manage data structures within a DB2 subsystem or between subsystems. This chapter introduces you to the interface for ALTER and CHANGE MANAGER, then steps you through the tasks that are necessary to create and migrate DB2 objects.

Before you use ALTER and CHANGE MANAGER, ensure that you have completed the tasks necessary to install the products. For information, see the installation guide.
Using the ALTER or CHANGE MANAGER interface

The ALTER and CHANGE MANAGER ISPF user interface consists of dialog panels that comply with Common User Access (CUA) conventions. To call or start ALTER or CHANGE MANAGER, you can use the BMC Software Administrative Products for DB2 panel that BMC provides (see Figure 6).

**NOTE**

If you have CHANGE MANAGER installed, ALTER is not an option on this panel.

---

**Figure 6  Administrative products for DB2 panel**

<table>
<thead>
<tr>
<th>COMMAND ===</th>
<th>BMC Software Administrative Products for DB2-</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CATALOG MANAGER for DB2 - Execute DDL or query DB2 Catalog</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CHANGE MANAGER for DB2 - Manage changes to DB2 objects/structures</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DASD MANAGER for DB2 - Manage or monitor DB2 physical objects</td>
<td></td>
</tr>
</tbody>
</table>

DB2 SSID . . . . . . DEAH (? = SSID List)
DB2 Catalog Access . . DIRECT (Direct)
Use Shared or Individual product ISPF APPLID? S (S/I - Admin Products only)

The BMC Software Administrative Products for DB2 panel contains the following features:

- the **COMMAND** line, on which you enter TSO commands
- the product selection list, which includes an input field with which you can specify an option from the list by entering a number
- the **DB2 SSID** field, in which you specify the DB2 subsystem (SSID) to be used
- the **DB2 Catalog Access** field, in which you specify whether to use the DB2 catalog data directly or to use a copy or a view of the DB2 catalog
the Use Shared or Individual product ISPF APPLID field, in which you specify whether to use a shared or individual ISPF profile ID

If you want to specify an ISPF Application ID (APPLID) for each product, use an individual APPLID. If you want to specify an APPLID for each DB2 subsystem, use a shared APPLID.

Using the Main Menu

After you select ALTER or CHANGE MANAGER from the Administrative Products for DB2 panel, the product’s Main Menu is displayed (see Figure 7).

Figure 7  CHANGE MANAGER Main Menu

You can use the ALTER Main Menu to select options to process work IDs, change installation options, perform specific tasks, and view a list of the latest enhancements to the product.

You can use the CHANGE MANAGER Main Menu to perform the same functions as ALTER, in addition to building CDL, processing baselines, using baseline and migrate profiles, and accessing the CM/PILOT component of CHANGE MANAGER.
Displaying the current environment

To display information about the current ALTER or CHANGE MANAGER environment, type ENV in the Command line on the Main Menu and press Enter. The BMC Environment panel is displayed, as shown in Figure 8.

Figure 8   BMC Environment panel

To obtain a list of the synonyms and tables that are dependent upon each plan, type SY in the Act column adjacent to the name of the plan.

To display information about fixes (PTFs) that you have applied to the product, type MAINT in the Command line on the BMC Environment panel and press Enter. The Product Maintenance List panel is displayed.

Using Fast Path Navigation

The Installation System for the Administrative products provides a feature called Fast Path Navigation. This feature enables you to switch from one product to another and then return to the original product. To initiate Fast Path Navigation, enter the name of the product to which you want to switch on the Command line of the current product. For a list of the products and commands, see Table 1.
For example, if you are currently using ALTER and want to view an object description in CATALOG MANAGER, enter BMCCAT on the ALTER Command line. The main menu for the requested product is displayed. In this case, the ALTER session is temporarily suspended and then resumed when you exit CATALOG MANAGER.

For information about enabling Fast Path Navigation, see the installation guide.

### Using ISPF commands

Most of the ISPF commands in ALTER and CHANGE MANAGER work the same as they do in other ISPF applications. Table 2 describes the most commonly used ISPF commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL</td>
<td>returns to the previous panel without saving any change that you made on the current panel</td>
</tr>
<tr>
<td>DOWN (or F8)</td>
<td>scrolls the panel down</td>
</tr>
<tr>
<td>END (or F3)</td>
<td>validates and processes information, the same as the Enter key</td>
</tr>
<tr>
<td>ENTER</td>
<td>processes information that is typed on the panel and executes any specified commands</td>
</tr>
<tr>
<td>HELP (or F1)</td>
<td>provides panel-level Help</td>
</tr>
</tbody>
</table>

### Fast Path Navigation commands

<table>
<thead>
<tr>
<th>Product</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>BMCALTER</td>
</tr>
<tr>
<td>CHANGE MANAGER</td>
<td>BMCCCHG</td>
</tr>
<tr>
<td>CATALOG MANAGER</td>
<td>BMCCAT</td>
</tr>
<tr>
<td>DASD MANAGER PLUS</td>
<td>BMCDASD</td>
</tr>
</tbody>
</table>
## Table 2  ISPF commands (part 2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| LEFT (or F10) | scrolls the panel to the left  
  **More:** < on a panel indicates that more information is available to the left.                                                                 |
| NEXT        | displays the next panel in a sequence                                                                                                                                                                      |
| PFSHOW      | displays the active function keys  
  Some panels use every available line to display input variables. To display all variables, enter `PFSHOW OFF` on the **COMMAND** line.                                                                 |
| PREV (or F12) | displays the previous panel in a sequence                                                                                                                                                                  |
| RIGHT (or F11) | scrolls the panel to the right  
  **More:** > on a panel indicates that more information is available to the right.                                                                                                 |
| SPLIT (or F2) | divides the panel and displays the ISPF Primary Option Menu in the new panel  
  If you start the product on both panels, ensure that each product is at the same version, release, and maintenance level.                                                                                 |
| SSE         | starts the BMC Simple Space Estimation (SSE) feature to estimate space requirements for table space or index objects                                                                                          |
| SWAP (or F9) | switches from one split panel to another                                                                                                                                                                    |
| UP (or F7)  | scrolls the panel up  
  **More:** - on a panel indicates that more information is available above the current line.                                                                                                           |
  
  Scrolling is available on a Model 2 3270 mainframe terminal, which uses a 24-line by 80-column display.                                                                                           |
| ZOOM (or F4) | displays the full value of an object with a long name in either a dialog or a panel  
  The cursor must be positioned on the long name value.                                                                                                                                       |
  
  On the following panels, you cannot use the **ZOOM** key to display the entire name of an object:                                                                                             |
  - Scope Rules panel (ACMFPMPSL)  
  - Change Rules panel (ALUFRULL)  
  - Help panels  
  For example, when you convert a table space from table-controlled partitioning to index-controlled partitioning, the products might display a Help panel that lists up to three partitioned indexes. If those indexes have long names, the product displays a portion of the names and disables the **ZOOM** key. |
Viewing BMC message information

The BMC Administrative products include informational, warning, and error messages. Most messages are assigned a message identifier (messageID) that consists of an alphabetic prefix and a number. Some message identifiers also include a severity code, such as W, E, or I. For information about viewing messages, see the Administrative Products for DB2 Messages Manual.

NOTE

Messages that are not assigned a message identifier display in the upper right-hand corner of the panel.

Achieving your goals with ALTER and CHANGE MANAGER

Before products like ALTER or CHANGE MANAGER were developed, changing the data structures in a DB2 environment was, at best, difficult. The process was also complex, tedious, time consuming, unreliable, and error prone. Whether working in a single DB2 subsystem or in a large, multiple-DB2 subsystem environment, the DBA must manage a large number of objects with complex interrelationships.

This section describes how ALTER and CHANGE MANAGER can automate and simplify the goals of changing data structures and migrating structure changes.

Creating, altering, or dropping DB2 objects

Creating, altering, or dropping DB2 objects in ALTER or CHANGE MANAGER is a three-stage process: specification, analysis, and execution.

- Specification

  Specify the changes that you want to make to the objects.

- Analysis

  Analyze the specified changes for accuracy and completeness. The result of the analysis stage is a worklist that contains the SQL and the information that is necessary to change the data structures and preserve all dependent objects.
**Execution**

Execute the worklist that is generated during the analysis stage.

---

**NOTE**

The changes that you specify in the specification stage and analyze in the analysis stage are actually made to the objects in the execution stage.

---

The following example illustrates these stages by showing you how to use the Task List feature of CHANGE MANAGER to quickly and easily change the length of a column in a table in your database.

Before you begin a set of tasks to achieve a particular goal, you should allocate your JCL data sets for analysis and execution, ensure that you are using the correct DB2 subsystem, and define your job card with the correct account information. For more information, see “Setting options” on page 163.

**To modify objects**

1. Start the Task List feature.

   The Task List feature of CHANGE MANAGER enables you to achieve various goals by taking you step-by-step through a series of tasks.

   A. On the CHANGE MANAGER Main Menu (see Figure 7), select Task List and press Enter.

   The Task List Menu is displayed, as shown in Figure 9.
B On the Task List Menu, select Create, alter, or drop DB2 objects and press Enter.

The Create, Alter, or Drop DB2 Objects panel is displayed, as shown in Figure 10.

Figure 9  Task List Menu

![Task List Menu](image)

Select a task. Then press Enter.
1 1. Create, alter, or drop DB2 objects
   2. Migrate DB2 objects using Specification
   3. Migrate DB2 objects using scope rules from a Migrate Profile

Commands: HELP END

Figure 10  Create, Alter, or Drop DB2 Objects panel

![Create, Alter, or Drop DB2 Objects panel](image)

Specify an Alter WORKID.
If the WORKID does not exist, the Create WORKID panel will be displayed.
To see a list of WORKIDs, type a wildcard pattern. (For example BM*.*)

Alter WORKID: RDACRJ.ALTER2

Type S to select processing options. Then press Enter to continue.

- Create or edit a WORKID
- Specify changes to DB2 objects
- Analyze the changes to create a worklist
- Execute the worklist to process the changes

Commands: HELP END
Specify an alter-type work ID.

A work ID enables you to identify, control, and track a unit of work. The alter-type work ID is used to store the changes to the DB2 objects that you specify.

A On the Create, Alter, or Drop DB2 Objects panel, type the name of an alter-type work ID, and type S to select all of the processing options. Then, press Enter.

The Create WORKID panel is displayed, as shown in Figure 11.

Figure 11  Create WORKID panel

```
ALUFWRKC -------------------------- Create WORKID ----------- WORKID NOT FOUND
Command ====>
WORKID . . . . . : RDACRJ.ALTER2
Type . . . . . . . 1 1. Alter
                   2. Migrate
                   3. Receive
Comment . . . . .

Select additional panels to display for Migrate type WORKID. Then press Enter.
  WORKID Migrate Options
  WORKID Change Rules

Commands:  HELP END CANCEL
```

B On the Create WORKID panel, select Alter for the Type. Then, press Enter.

The Object Specification panel is displayed, as shown in Figure 12.
Figure 12  Object Specification panel

ALUSOS TER  ---------------------- Object Specification ----------------------
Command ===>
WORKID ... : RDACRJ.ALTER2  _ Changed Objects List
Specify the object type(s) to be included in a list.

Stogroup .  (SG   Name)
Database .  (DB   Name)
Tablespace  (DB.TS Name)
Table      (Own.TB Name)
Check Cnst. (Own.TB.Chk)
Foreign Key (Own.TB.Rel)
Index      (Own.IX Name)
Unique Cnst (Own.TB.Uniq)
View       (Own.VW Name)
Trigger    (Sch.TR Name)
Alias      (Own.AL Name)
Synonym    (Own.SY Name)

The following objects are retrieved independently of all other DB2 objects.
Sequence   (Sch.SQ Name)

3 Specify the changes that you want to make to the objects.

The current values of the objects are displayed on panels. You can modify the attributes for objects by specifying new values or by typing new values over the existing values. All of your changes are stored in DB2 tables.

A On the Object Specification panel, type the name of your database, table space, and table. Then, press Enter.

The Mixed List panel is displayed, as shown in Figure 13. From this panel, you begin the process of specifying changes to the objects that are listed.
Creating, altering, or dropping DB2 objects

Figure 13  Mixed List panel

On the Mixed List panel, in the Act field type E next to the table that you want to edit, and press Enter.

The Table Detail panel is displayed, as shown in Figure 14.

Figure 14  Table Detail panel
C On the Table Detail panel, type S to select the Table Column List, and press Enter.

The Table Columns List panel is displayed, as shown in Figure 15. This panel displays the current column attributes for the table. In this example, you will change the length of a column.

Figure 15  Table Columns List panel

D On the Table Columns List panel, tab to the Length field for the column that you are modifying. Type a new value for the length of the column over the existing value. Press Enter.

E Press END to return to the Table Detail panel. Press END again to return to the Mixed List Panel.

The Mixed List panel, shown in Figure 16, displays an “*A” next to the table, indicating that you have made a change to the table.
Press END to return to the Object Specification panel. Press END again.

The Analysis Alter Worklist Interface panel is displayed, as shown in Figure 17.

Figure 16  Mixed List panel showing changes

Figure 17  Analysis Alter Worklist Interface panel
4 Analyze the changes that you specified.

The Analysis component of the product examines the changes that you specified to determine the effect that the changes have on the structure and the related or dependent objects.

A On the Analysis Alter Worklist Interface panel, select the option to **Generate a worklist which will convert and reload current data**. Then select to run the analysis in **Foreground**. Press Enter.

---

**NOTE**

When you run Analysis in foreground, your TSO session will be unavailable until the processing has completed.

---

The Analysis JCL Processing Interface panel is displayed, as shown in **Figure 18**.

**Figure 18 Analysis JCL Processing Interface panel**

ACMFANL3 -------------- Analysis JCL Processing Interface --------------
Command ==> WORKID . . . . : RDACRJ.ALTER2

Specify Dataset Names
- Worklist . . . . 'RDACRJ.V10.WLBASE01(ALTER2)'
- Diagnostics . . . . SYSOUT
- Sysout Class . . . X (Used only when Diagnostics = SYSOUT)

Select foreground processing options. Then press Enter.
- Override
- S Create analysis input
- S Edit analysis input
- S Run analysis
- S Edit Worklist

Commands: BROWSE PREVIOUS HELP END

B On the Analysis JCL Processing Interface panel, type the name of your **Worklist**, and type **S** to select your processing options. Press **Enter**.

The input for the Analysis process is displayed, as shown in **Figure 19**. The input includes the ALUIN input stream, which provides keywords for analyzing the changes. For more information about the input stream, see the **ALTER and CHANGE MANAGER for DB2 Reference Manual**.
Creating, altering, or dropping DB2 objects

Figure 19  Analysis input

<table>
<thead>
<tr>
<th>ISREDDE2</th>
<th>SYS11042.T172413.RA000.RDACRJ.WLIN.H03</th>
<th>Columns 00001 00072</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ====&gt; PAGE</td>
<td></td>
</tr>
<tr>
<td>Top of Data</td>
<td>Bottom of Data</td>
<td></td>
</tr>
</tbody>
</table>

C  Press F8 to scroll down the input. Press END.

D  Press Enter to run Analysis.

The results of Analysis are a diagnostic output file (ALUPRINT) and a worklist.

E  To review the results of Analysis (ALUPRINT), you can split your ISPF session or you can use your normal method to review SYSOUT. The last message in the diagnostic output (Figure 20) indicates that the worklist completed successfully.
Figure 20  Analysis diagnostic output

Figure 21 shows the worklist that was written to the data set that you specified on the Analysis JCL Processing Interface panel.

Figure 21  CHANGE MANAGER alter-type worklist (part 1 of 3)
**Figure 21** CHANGE MANAGER alter-type worklist (part 2 of 3)

```
* SORTDEVT SYSDA
* INCLUDE (DATA AMS SQL REBIND )
* NOREGENIDENTITY
* SMSINCLUDE ()
* NODBRMLIB ORDERBY
* NOVVALPROP
* UNLOADEMPTY NOSTOPCOMMIT TABLEACCESS
* NOREORGALT
* BMCCHECK IBMREORG REORGONLINE REORGALL REORGREF
* BCMREBUILD
* SYCNPINT 10
* STANDALONE STATS ALLSTATSUPD TABLEALL NOHISTORY
* BMCCOPY IBMLOAD IBMUNLOAD
* DYNUNLD
* DYNCOPY
* NOPARTCOPY
* COPYDDN(COPY01 )
* NOUTILCOPY
* NOPARALLEL

***********************************************************************

-TIME 000050 '2011-02-14-13.39.18.734687'
-SSID 000100 DEGA
-WKID 000150 RDACRJ.ALTER2

***********************************************************************

-SYNC 000200 START OF EXECUTION SYCNPINT

***********************************************************************

-SQL 000250 ALTER TABLE JIBASIC.T_T01AICP
    ALTER COLUMN COLC_1
    SET DATA TYPE CHAR(40)

-SYNC 000300 END OF ALTER TABLE SECTION

***********************************************************************

-REOR 000350
    REORG TABLESPACE JIBASIC.T01AICP
    UNLDDN SYSR1001
    COPYDDN (SYCL0001)
    SORTKEYS
    SORRTDATA LOG NO
    SHRLEVEL CHANGE
    MAXRO 0
    MAPPINGTABLE MAP_TABLE

-SYNC 000400 REORG OF TABLESPACE JIBASIC.T01AICP COMPLETE
-SYNC 000450 END OF REORG TABLESPACE SECTION

***********************************************************************

-BMCS 000500
    BMCSTATS TABLESPACE JIBASIC.T01AICP
    TABLE ALL
    UPDATEDB2STATS YES
    KEYCARD Y
    REPORT NO

```
Press END.

The Execution JCL Build Interface panel is displayed, as shown in Figure 22.

Figure 22  Execution JCL Build Interface panel

```
ACMFEXCO ------------------ Execution JCL Build Interface ------------------
Command ===>

WORKID . . . . : RDACRJ.ALTER2

Select JCL and run type. Then press Enter.
JCL Type . . 1 1. Build Initial JCL
   2. Build Restart JCL from previous execution JCL
   3. Build Startover JCL from previous execution JCL

Run Type . . 1 1. Build JCL in Foreground
   2. Build JCL in Batch

Commands: NEXT HELP END
```
5 Execute the changes that you specified and analyzed.

The Execution component of the product uses the worklist as its input job stream and performs the tasks contained in the worklist. The worklist acts as a detailed procedure for implementing the changes that you defined in the specification stage.

A On the Execution JCL Build Interface panel, press Enter to accept the defaults.

The Execution Pre- and Post-Processing Interface panel is displayed, as shown in Figure 23.

**Figure 23 Execution Pre- and Post-Processing Interface Panel**

```
ACMFEXC1 ---------- Execution Pre- and Post-Processing Interface ---------------
Command ===>
WORKID . . . . : RDACRJ.ALTER2
Specify additional steps to be included in this job. Then press Enter.
Pre-Execution Compare
  1. Do not do a compare before execution
  2. Compare two previous baselines
  3. Compare a previous baseline to the current catalog
Pre-Execution Baseline
  1. Do not build a baseline before execution
  2. Build a baseline before execution
Post-Execution Compare
  1. Do not do a compare after execution
  2. Compare a previous baseline to the new catalog structures
  3. Compare the baseline built during a previous job step to the new catalog structures
  4. Create CDL to fall back to a previous baseline
Post-Execution Baseline
  1. Do not build a baseline after execution
  2. Build a baseline after execution
Commands: HELP PREVIOUS END
```

B On the Execution Pre- and Post-Processing Interface Panel, press Enter to accept the defaults.

The Execution JCL Processing Interface panel is displayed, as shown in Figure 24.
C On the Execution JCL Processing Interface panel, specify the data set name for the Execution JCL, accept the defaults for the JCL build options, and type S to select the processing options. Press Enter.

The Execution JCL is displayed, as shown in Figure 25. The JCL includes the //AEXIN DD statement (input stream), which includes keywords that are used in executing the JCL. For more information about the input stream, see the ALTER and CHANGE MANAGER for DB2 Reference Manual.

Figure 24 Execution JCL Processing Interface panel

Figure 25 Execution JCL for an alter-type worklist (part 1 of 4)
Figure 25  Execution JCL for an alter-type worklist (part 2 of 4)

```
// COND=(4,LT)
// STEPLIB DD DSN=SYS3.DEFF.DSNEXIT,DISP=SHR
//      DD DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
//      DD DISP=SHR,DSN=ACM.SMP1010.DBLINK
//      DD DISP=SHR,DSN=CSGI.SASC.V700C.LINKLIB
//      DD DISP=SHR,DSN=CSGI.SASC.V700C.LOAD
// * ************************************************************
// ABNLIGNR DD DUMMY
// SYSUDUMP DD SYSOUT=*  
// SYSTERM DD SYSOUT=*  
// ASUSRPRRT DD SYSOUT=*  
// UTPRINT DD SYSOUT=*  
// *--------------------------------------------------------------------
// *  STATISTICS WORK DD STATEMENTS
// *--------------------------------------------------------------------
// STPRIN01 DD SYSOUT=*  
// RNPRIN01  DD SYSOUT=*  
// SYSPRINT DD SYSOUT=*  
// SYSIN    DD SPACE=(CYL,(15,15)),UNIT=SYSDA,DISP=(NEW,DELETE),
//             DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB
// SYSTSIN  DD SPACE=(TRK,(1,1)),UNIT=SYSDA,DISP=(NEW,DELETE),
//             DSORG=PS,LRECL=80,BLKSIZE=3200,RECFM=FB
// SYSPRINT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,
//             DISP=(NEW,DELETE)
// SYSTSPRT DD SPACE=(CYL,(15,15)),UNIT=SYSDA,
//             DISP=(NEW,DELETE)
// ALUIN    DD SPACE=(CYL,(5,5)),UNIT=SYSDA,DISP=(NEW,DELETE),
//             DSORG=PS,LRECL=137,BLKSZ=3155,RECFM=VBA
// ALUPRINT DD SPACE=(TRK,(1,1)),UNIT=SYSDA,DISP=(NEW,DELETE),
//             DSORG=PS,LRECL=80,BLKSZ=3200,RECFM=FB
// ALJXPRINT DD SPACE=(CYL,(5,5)),UNIT=SYSDA,DISP=(NEW,DELETE)
// *** RUNTIME ENABLEMENT CHosen
// SYSPROC  DD DSN=AEX.QA1010.BMCCLIB,DISP=SHR
// ACT$MSGS DD DSN=AEX.QA1010.BMCMLIB,DISP=SHR
// ISPMLIB  DD DSN=AEX.QA1010.BMCMLIB,DISP=SHR
// ** ----------------------------------------------------------------
// **     SYSTEM ISPF MLIB       SKELETON AJXISPFS
// ** ----------------------------------------------------------------
// DD   DSN=ISP.SISPMENU,DISP=SHR
// ***************************************************
// ISPSLIB  DD DSN=AEX.QA1010.BMCSLIB,DISP=SHR
// ISPTLIB  DD DSN=AEX.QA1010.BMCSTLIB,DISP=SHR
```
Figure 25  Execution JCL for an alter-type worklist (part 3 of 4)
After you review the JCL, press END to return to the Execution JCL Processing Interface panel.

From the Execution JCL Processing Interface panel, press Enter to submit the JCL and process your changes.

The result of execution is a worklist execution log (AEXPRINT).

To review the results of execution in the worklist execution log (AEXPRINT), you can split your ISPF session or you can use your normal method to review SYSOUT. Figure 26 shows the worklist execution log.

The last message in the diagnostic output indicates that the work ID status is complete and that the worklist completed successfully. The worklist implemented the structure changes that you specified.
Figure 26  Execution results for an alter-type worklist (part 1 of 6)

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-14 14.14.05 ON SYSID DB2B WORKLIST EXECUTION LOG PAGE 1

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

BMC706707I PTFs:
BMC16815I PARAMETERS: ACM
BMC16815I PARAMETERS: SSID DEGA WORKID RDACRJ.ALTER2
BMC16815I PARAMETERS: DYNWORKUNIT SYSDA
BMC16815I PARAMETERS: DASDDOPT DS10SRGA
BMC16815I PARAMETERS: CATDOPT DC101DOP
BMC16815I PARAMETERS: STOPWAIT 3
BMC16815I PARAMETERS: STOPWTSSECS 10
BMC16815I PARAMETERS: HASHWARNRC 04
BMC16815I PARAMETERS: LOCK EXCLUSIVE
BMC16992I USING 'DS10SRGA' AS OPTIONS FOR DASD MANAGER UTILITIES
BMC16943I USING 'DMAISEGA' FOR RUN OPTIONS
BMC16971I WORKLIST DSNAME IS 'RDACRJ.V9.WLBASE01(ALTER2)'
BMC16998I DEBUG: DB2 RELEASE: 999
BMC16898I PARAMETERS ACCEPTED, BEGINNING EXECUTION VERSION V10.01.00 03/24/2011 FOR BMC CHANGE MANAGER
BMC16911 AUTHID SWITCHING OPTION 'N' IS IN EFFECT. -AUTH NOT ALLOWED.
BMC16957I EXECUTION AUTHID IS RDACRJ

:*** CHANGE MANAGER FOR DB2 VERSION 10.01.00(04/08/2011)
BMC70933I WORKID DESCRIPTION:
BMC16991I INITIAL WORKID STATUS FOR RDACRJ.ALTER2 IS 'N' (NOT STARTED
BMC16999I UPDATED WORKID STATUS FOR RDACRJ.ALTER2 IS 'S' (STARTED)
:*** CHANGE MANAGER FOR DB2 VERSION 10.01.00(04/08/2011)
:***********************************************************************
:*** PTFS APPLIED TO PRODUCT:
:*** NONE
:***********************************************************************
:* ALUIN PARMS:
:* SSID DEGA
:* WORKID RDACRJ.ALTER2
:* SORTDEV SYSDA
:* INCLUDE (DATA AMS SQL REBIND )
:* NOREGENIDENTITY
:* SMSINCLUDE ()
:* NODBRMLIB ORDERBY

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-14 14.14.05 ON SYSID DB2B WORKLIST EXECUTION LOG PAGE 2

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

:* NOVVALPROP
:* UNLOADEMPTY NOSTOPCOMMIT TABLEACCESS
:* NOREORGALT
:* BMCCHECK IBMREORG REORGONLINE REORGALL REORGREF
:* BMCREBUILD
:* SYNCPPOINT 10
:* STANDALONESTATS ALLSTATSUPD TABLEALL NOHISTORY
:* BMCCOPY IBMLOAD IBMUNLOAD
:* DYNURLD
:* DYNCOPY
:* NOPARTCOPY
:* COPYON(COPY01 )
:* NOUTILCOPY
:* NOPARALLEL
:***********************************************************************
Figure 26  Execution results for an alter-type worklist (part 2 of 6)

```sql
BEGIN ALTER TABLE J1BASIC.T_T01AICP
ALTER COLUMN COLC_1
SET DATA TYPE CHAR(40)
END ALTER TABLE SECTION
```
**Figure 26 Execution results for an alter-type worklist (part 3 of 6)**

<table>
<thead>
<tr>
<th>Message ID</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNU303I</td>
<td>14:14:00.74</td>
<td>(RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=1 FOR TABLE J1BASIC.T_T01AICP PART=4</td>
</tr>
<tr>
<td>DSNU304I</td>
<td>14:14:00.74</td>
<td>(RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=2 FOR TABLE J1BASIC.T_T01AICP</td>
</tr>
<tr>
<td>DSNU302I</td>
<td>14:14:00.79</td>
<td>(RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:00</td>
</tr>
<tr>
<td>DSNU300I</td>
<td>14:14:00.82</td>
<td>SORT PHASE STATISTICS - NUMBER OF RECORDS=4 ELAPSED TIME=00:00:00</td>
</tr>
<tr>
<td>DSNU548I</td>
<td>14:14:01.15</td>
<td>BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX J1BASIC.I_T01AICP PART</td>
</tr>
<tr>
<td>DSNU549I</td>
<td>14:14:01.28</td>
<td>BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX J1BASIC.I_T01AICP PART</td>
</tr>
<tr>
<td>DSNU580I</td>
<td>14:14:01.39</td>
<td>BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX JEFF.JFLIXMAP</td>
</tr>
<tr>
<td>DSNU591I</td>
<td>14:14:01.39</td>
<td>BUILD PHASE COMPLETE, ELAPSED TIME=00:00:00</td>
</tr>
<tr>
<td>DSNU592I</td>
<td>14:14:01.99</td>
<td>COPY PROCESSED FOR TABLESPACE J1BASIC.T01AICP NUMBER OF PAGES=20 AVERAGE PERCENT FREE SPACE PER PAGE = 9.40 PERCENT OF CHANGED PAGES = 100.00 ELAPSED TIME=00:00:02</td>
</tr>
<tr>
<td>DSNU893I</td>
<td>14:14:03.36</td>
<td>SWITCH PHASE COMPLETE, ELAPSED TIME = 00:00:01</td>
</tr>
<tr>
<td>DSNU894I</td>
<td>14:14:03.36</td>
<td>DB2 IMAGE COPY SUCCESSFUL FOR TABLESPACE J1BASIC.T01AICP</td>
</tr>
<tr>
<td>DSNU905I</td>
<td>14:14:05.27</td>
<td>UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0</td>
</tr>
</tbody>
</table>

--

2011-02-14 14.14.05 ON SYSID DB28 WORKLIST EXECUTION LOG PAGE 5

**TABLE ALL**
**UPDATEDB2STATS YES**
**KEYCARD Y**
**REPORT NO**
**50944631**

BMC16815I PARAMETERS: DEGA,...,DSNRSIGA,ALTER2,ALTER2
BMC16822I CONTROL PASSED TO 'ASUSMAIN'
BMC16600 DEBUG: ASUSMAIN 0

BMC166600 ***** D A S D M A N A G E R V10.01.00 F O R D B 2 *****
BMC16674 ***** D B 2 V1010 (V101) *****

BMC2781181 DB2 SERVICES MANAGER V10.01.00
BMC278113I MAINT: ASUMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: ASMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: AEXMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: SCCMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: BMUMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: ACSMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: ATSMAINT
BMC278113I MAINT: NONE
BMC278113I MAINT: D2UMAINT
Figure 26  Execution results for an alter-type worklist (part 4 of 6)
Figure 26  Execution results for an alter-type worklist (part 5 of 6)

```sql
REOPT  NONE
KEEPDYNAMIC  NO
IMMEDWRITE  NO
DBPROTOCOL  DRDA
OPTHINT
ENCODING  EBCDIC(00000)
PLANMGMT  OFF
PLANMGTSCOPESCOPE  STATIC
CONCURRENTACCESSRESOLUTION

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-14 14.14.15 ON SYSID DB2B
WORKLIST EXECUTION LOG PAGE 7

----------------------------------------------------------------------------------------------------------------
LINE PREFIX: DENOTES WORKLIST INPUT STATEMENTS
----------------------------------------------------------------------------------------------------------------
EXTENDEDINDICATOR  NO
PATH  "SYSIBM","SYSFUN","SYSPROC","RDAJZB"

DSNT232I  *DEGA SUCCESSFUL REBIND FOR
PACKAGE = DEGA.JIBASIC.TROIICP2().

DSN END
*:REBD 00070D
: REBIND TRIGGER PACKAGE ( -
:  JIBASIC.TROIICP1 +
:  )
: PLANMGMT(OFF) -
: EXPLAIN(NO)  13257620
BMC16822I  CONTROL PASSED TO 'IKJEFT01'
BMC16898I  DEBUG: IKJEFT01 0

READY
DSN SYSTEM(DEGA)

DSN7254I  *DEGA DSNTBB2 REBIND OPTIONS FOR
PACKAGE = DEGA.JIBASIC.TROIICP1().
ACTION
OWNER  RDAJZB
QUALIFIER  RDAJZB
VALIDATE  BIND
EXPLAIN  NO
ISOLATION  CS
RELEASE  COMMIT
COPY
APREUSE
APCOMPARE
APRETAINDUP  YES

DSNT255I  *DEGA DSNTBB2 REBIND OPTIONS FOR
PACKAGE = DEGA.JIBASIC.TROIICP1().
SQLERROR  NOPACKAGE
CURRENTDATA  NO
DEGREE  1
DYNAMICRULES  BIND
NODEFER  PREPARE
REOPT  NONE
KEEPDYNAMIC  NO
IMMEDWRITE  NO
DBPROTOCOL  DRDA
OPTHINT
ENCODING  EBCDIC(00000)
PLANMGMT  OFF
PLANMGTSCOPESCOPE  STATIC

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00

```
Migrating DB2 objects

Just as changing DB2 objects is a three-stage process, migrating DB2 objects from one environment to another within the same DB2 subsystem in ALTER or CHANGE MANAGER is also a three-stage process. The three stages are specification, analysis, and execution.

- **Specification**
  
  Identify the objects to be migrated and specify changes to the data structures.

- **Analysis**
  
  Analyze the specified migration changes for accuracy and completeness. The result of the analysis stage is a worklist that contains the SQL and the information that is necessary to generate the data structure in the target DB2 system.
Execution

Execute the worklist that is generated during the analysis stage.

**NOTE**
The changes that you specify in the specification stage and analyze in the analysis stage are actually made to the objects in the execution stage.

The following example illustrates these three stages by showing how to use the Task List feature of CHANGE MANAGER to quickly and easily migrate objects from one environment to another environment within the same DB2 subsystem.

**NOTE**
Remember that CHANGE MANAGER can also migrate only the changes that you make to a data structure on one DB2 subsystem to other DB2 subsystems.

Before you begin a set of tasks to achieve a particular goal, you should allocate your JCL data sets for analysis and execution, ensure that you are using the correct DB2 subsystem, and define your job card with the correct account information.

**To migrate objects**

1. Start the Task List feature.

   The Task List feature of CHANGE MANAGER enables you to achieve various goals by taking you step-by-step through a series of tasks.

   **A** On the CHANGE MANAGER Main Menu (see Figure 7), select Task List and press Enter.

   The Task List Menu is displayed, as shown in Figure 27.

The Migrate DB2 Objects Using Specification panel is displayed, as shown in Figure 28.

Figure 27  Task List Menu

![Task List Menu]

Figure 28  Migrate DB2 Objects Using Specification panel

![Migrate DB2 Objects Using Specification panel]
2 Specify a migrate-type work ID.

A work ID enables you to identify, control, and track a unit of work. The migrate-type work ID is used to specify data structure definitions to be migrated from one DB2 subsystem to another, or to duplicate structures within the same subsystem.

A On the Migrate DB2 Objects Using Specification panel, type the name of a migrate-type work ID, and type S to select all of the processing options except Create or edit a Migrate Profile. Then, press Enter.

The Create WORKID panel is displayed, as shown in Figure 29.

Figure 29 Create WORKID panel

B On the Create WORKID panel, select Migrate for the Type. Type S to select the WORKID Change Rules. Then, press Enter.

The Change Rules panel is displayed, as shown in Figure 30.
Change rules enable you to specify global changes to the name, owner, creator, or other attributes of the objects. The new attributes that you specify apply to all objects in the migration that possess the attributes.

C On the Change Rules panel, type E in the Act field to select Edit. Press Enter.

The Edit Change Rules panel is displayed, as shown in Figure 31.
On the Edit Change Rules panel, select Change as the Option for the attribute, select All as the Object, and type S to select Display Attribute List. Press Enter.

By selecting All objects, you can specify one change rule that changes an attribute for all of the tables, indexes, views, and so on. You can change an attribute that applies to all of the migrated objects that have that attribute.

The Edit Rules Attribute Selection panel is displayed, as shown in Figure 32.
E On the Edit Rules Attribute Selection panel, select the number for the Name attribute. Press Enter, and then press END.

The Edit Change Rules panel is displayed, as shown in Figure 33.

Figure 32 Edit Rules Attribute Selection panel

Figure 33 Edit Change Rules panel—NAME attribute
**F** On the Edit Change Rules panel, type the **Current Attribute Value** and the **New Attribute Value** for the database name. Press END.

The change rules are displayed in the Change Rules panel, as shown in Figure 34.

**Figure 34  Change Rules panel—NAME attribute**

On the Change Rules panel, type **L** in the **Act** field for the name rule. The **L** enables you to create a new rule like the existing rule. Press **Enter**.

**H** On the second line, type **OWNER** over **NAME** in the **Attribute** field, type * for the **Current Attribute Value** (if you do not know the current owner), and type your TSO logon ID (or other desired value) for the **New Attribute Value**. Press **Enter**.

The Change Rules panel is displayed, as shown in Figure 35.
I Press END.

The Edit WORKID Migration Options panel is displayed.

J Press END to return to the Object Specification panel, as shown in Figure 36.

Figure 35  Change Rules panel—OWNER attribute

<table>
<thead>
<tr>
<th>ALUFRULL ------------------</th>
<th>Change Rules -------------------------</th>
<th>SCROLL ====&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORKID . . . : RDACRJ.MIGRATE2</td>
<td>Type action next to existing rule or overtype existing values.</td>
<td></td>
</tr>
<tr>
<td>E=Edit I=Insert D=Delete L=Like C=Create</td>
<td>Rules 1 to 2 of 2</td>
<td></td>
</tr>
<tr>
<td>Act Opt Obj Attribute Current Attribute Value New Attribute Value</td>
<td>More: &gt;</td>
<td></td>
</tr>
<tr>
<td>***************************************** TOP *****************************************</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C ** NAME J1BASIC J2BASIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C ** OWNER * RDACRJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>***************************************** BOTTOM *****************************************</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 36  Object Specification panel

<table>
<thead>
<tr>
<th>ALUSOS GRATE ----------------------</th>
<th>Object Specification ----------------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt; WORKID . . . : RDACRJ.MIGRATE2</td>
<td>Specify the object type(s) to be included in a list.</td>
</tr>
<tr>
<td>Stogroup . (SG Name)</td>
<td>(DB Name)</td>
</tr>
<tr>
<td>Database. (DB.Name)</td>
<td>Table . (Own.TB Name)</td>
</tr>
<tr>
<td>Tablespace (DB.TS Name)</td>
<td>Check Cnst. (Own.TB.Cnk)</td>
</tr>
<tr>
<td>Table . (DB.TB Name)</td>
<td>Foreign Key (Own.TB.Rel)</td>
</tr>
<tr>
<td>Check Cnst. (Own.TB.Cnk)</td>
<td>Index . (Own.IX Name)</td>
</tr>
<tr>
<td>Foreign Key (Own.TB.Rel)</td>
<td>Unique Cnst (Own.TB.Uniq)</td>
</tr>
<tr>
<td>Index . (Own.IX Name)</td>
<td>View . . . (Own.VW Name)</td>
</tr>
<tr>
<td>Unique Cnst (Own.TB.Uniq)</td>
<td>Trigger . . (Sch.TR Name)</td>
</tr>
<tr>
<td>View . . . (Own.VW Name)</td>
<td>Alias . . (Own.AL Name)</td>
</tr>
<tr>
<td>Trigger . . (Sch.TR Name)</td>
<td>Synonym . . (Own.SY Name)</td>
</tr>
<tr>
<td>Alias . . (Own.AL Name)</td>
<td>Synonym . . (Own.SY Name)</td>
</tr>
</tbody>
</table>

The following objects are retrieved independently of all other DB2 objects.

Sequence . (Sch.SQ Name)

Stored Proc (Sch.SP Name)
3 Identify the objects and dependent objects that you want to migrate.

A On the Object Specification panel, type the name of a **Database**. Press **Enter**.

The Mixed List panel is displayed, as shown in **Figure 37**.

**Figure 37**  **Mixed List panel**

```
ALUSMXL RATE --------------------------- Mixed List ----------------------------
Command ===>                                                    Scroll. . CSR
WORKID . . . : RDACRJ.MIGRATE2                                      Commands: CANCEL

Type action next to object and press Enter.
E=Edit  L=Like  D=Drop  U=Undo  M=Migrate  MO=Migrate options
            Objects 1 to 1 of 1
            More:

Act     Object-Type    Objects
************************************ TOP **************************************
DB . . . . . J1BASIC  *********************************** BOTTOM ********
```

B On the Mixed List panel, type Z in the **Act** field next to the database. Press **Enter**.

The Mixed List panel is displayed with the dependent objects of the database, as shown in **Figure 38**.
C On the Mixed List panel, type M in the Act field next to the database that you want to migrate. Press Enter.

D Type MO in the Act field next to the database that you selected. Press Enter.

The Migrate Options Overrides panel is displayed, as shown in Figure 39.
On the Migrate Options Overrides panel, type **Y** to select the dependent objects that you want to migrate. Press **Enter**, and then press **END**.

In this example, data was selected. No dependent objects were selected.

The Mixed List panel is displayed, as shown in **Figure 40**. An “*M*” is displayed next to the database that you selected to migrate.
Figure 40  Mixed List panel—database selected for migration

F  On the Mixed List panel, type M in the Act field next to the table space that you want to migrate. Press Enter.

G  Type MO in the Act field next to the table space that you selected. Press Enter. The Migrate Options Overrides panel is displayed.

H  On the Migrate Options Overrides panel, type Y to select the dependent objects that you want to migrate. Press Enter, and then press END.

In this example, data and all of the dependent objects were selected.

The Mixed List panel is displayed, as shown in Figure 41. An “*M” is displayed next to the table space that you selected to migrate.
Press END to return to the Object Specification panel. Press END again.

The Analysis Migrate Worklist Interface panel is displayed, as shown in Figure 42.

**Figure 41 Mixed List panel—objects selected for migration**

```
ALUSMXL RATE --------------------------- Mixed List ---------------------------
Command ===> Scroll . CSR
WORKID . . . . : RDACRJ.MIGRATE2 Commands: CANCEL

Type action next to object and press Enter.
E=Edit  L=Like  D=Drop  U=Undo  M=Migrate  MO=Migrate options
Objects 1 to 13 of 53
More: +

Act Object-Type   Objects
*************************** TOP ***************************
*M  DB . . . . . . J1BASIC
   TSG . . . . J1BASIC T_T10AIM
   TB . . . . J1BASIC T_T10AIM
   IX . . . J1BASIC I_T10AIM1
*M  TSI . . . . . J1BASIC T01AICP
   TB . . . . J1BASIC T_T01AICP
   IXC . . . J1BASIC I_T01AICP1
   #TR . . . J1BASIC TR01ICP1
   #TR . . . J1BASIC TR01ICP2
   TS . . . . J1BASIC T02ASSEG
   TB . . . . J1BASIC T02ASSEG
   IX . . J1BASIC I_T02ASSEG_P
   IX . . J1BASIC I_T02ASSEG_X

ACMFANL2 --------------- Analysis Migrate Worklist Interface ---------------
Command ===> WORKID . . . . : RDACRJ.MIGRATE2
Analyze "Migrate" WORKID

Select type of analysis.
1  1. Analyze Specification entries (use the WORKID's change rules)
2. Analyze Specification entries (use a Migrate Profile's locations and change rules)
3. Analyze Migrate Profile scope (use the Migrate Profile's locations and change rules)

Type an Outbound Migrate Profile, or type a wildcard pattern for a selection list if analysis type is 2 or 3.
Migrate Profile . .

Select run type. Then press Enter.
Run Type . . . . 2 1. Foreground
2. Batch

Commands: HELP END
```
4 Analyze the migration changes that you specified.

The Analysis component of the product examines the changes that you specified and generates a worklist that contains the instructions that are necessary to implement the actions chosen during the specification stage.

A On the Analysis Migrate Worklist Interface panel, select the Analyze Specification entries (use the WORKID's change rules) option. Then select to run the Analysis in Batch. Press Enter.

--- NOTE ---
If you run Analysis in foreground, your TSO session will be unavailable until the processing has completed.

The Analysis JCL Processing Interface panel is displayed, as shown in Figure 43.

**Figure 43  Analysis JCL Processing Interface panel**

```
ACMFANL3 -------------- Analysis JCL Processing Interface --------------
Command   ===>  
WORKID . . . : RDACRJ.MIGRATE2

Specify Dataset Names
  JCL . . . . : 'RDACRJ.V10.ANLYJCL(MIGRATE2)'
  Worklist . . . : 'RDACRJ.V10.WLBASE01(MIGRATE2)'
  Diagnostics . . . : SYSOUT

Select batch processing options. Then press Enter.
  S Override
  S Create JCL
  S Edit JCL
  S Submit JCL
  S Edit Worklist
Commands:  BROWSE PREVIOUS HELP END
```

B On the Analysis JCL Processing Interface panel, specify your data set names, and type S to select the processing options. In this example, one of the options for analysis is overridden. Press Enter.

The Analysis Worklist Options panel is displayed, as shown in Figure 44.
On the Analysis Worklist Options panel, press Enter to accept the defaults.

The Analysis Worklist Command Options panel is displayed, as shown in Figure 45.

**Figure 44  Analysis Worklist Options panel**

```
ALUFOAN1 ----------------- Analysis Worklist Options --------------------------
Command ===> 
Type information. Then press Enter to continue.

Specify allocation parameters for new datasets.
  Dataset    Unitname   Volume   Priqty   Secqty   Alloc Unit
  Worklist   SYSDA 15 5   Tracks

Specify optional Global AUTHID.
  Global AUTHID . .

Specify sync point frequency.
  Frequency . . . 10

Commands: HELP END
```

**Figure 45  Analysis Worklist Command Options panel**

```
ALUFOAN2 ---------------- Analysis Worklist Command Options ---------------------
Command ===> 
Type information. Then press Enter to continue or PF12 for previous panel.

Select the elements to include in this worklist.
  S SQL
  S Data
  S AMS
    _ STORCLAS
    _ MGMTCLAS
    _ DATACLAS
    _ Bind
    _ DBRMLIB
    _ Bindauth

Commands: HELP END PREVIOUS
```
Migrating DB2 objects

D On the Analysis Worklist Command Options panel, press Enter to accept the defaults.

The Analysis Input Stream Parameters panel is displayed, as shown in Figure 46.

Figure 46 Analysis Input Stream Parameters panel

E On the Analysis Input Stream Parameters panel, type S to select the option Omit the -STOP command between phases of migrate worklist. When you migrate within a single subsystem, you normally do not want to stop between phases in a migrate worklist. (For more information, see the ALTER and CHANGE MANAGER for DB2 User Guide.)

F Press END.

The JCL Generation component is invoked and Batch Analysis JCL Generation is displayed, as shown in Figure 47. The JCL includes the //ALUIN DD and //AJXPOFIN DD statements (input streams), which include keywords for processing the worklist. For more information about the input streams, see the ALTER and CHANGE MANAGER for DB2 Reference Manual.
Figure 47  Batch Analysis JCL (part 2 of 7)

```plaintext
//***************************************************
//*  CREATED BY :  RDACRJ
//*  TIMESTAMP  :  02/14/2011.16.02.51
//*  ENVIRONMENT:  ISPF 6.1MVS  TSO
//*  RELEASE    :  V10.01.00 04/08/2011
//*  DB2 VERSION:  1000
//***************************************************

****---------------------------------------------------------------------
****        BMC CHANGE MANAGER ANALYSIS
****---------------------------------------------------------------------

//ANALYSIS EXEC PGM=ALUAMAIN,
  //  REGION=0M,
  //  PARM='DMA1SEGA'
//STEPLIB DD  DSN=SYS3.DEFF.DSNEXIT,DISP=SHR
  //  DD  DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
  //  DD DISP=SHR,DSN=ACM.SMP1010.DBLINK
  //  DD DISP=SHR,DSN=CSGI.SASC.V700C.LINKLIB
  //  DD DISP=SHR,DSN=CSGI.SASC.V700C.LOAD
// ** ************************************************************************
//ABNLIGNR DD DUMMY
//SYSUDUMP DD SYSOUT=* 
//ALUPRINT DD SYSOUT=* 
//AJXPOFVL DD DUMMY
//ALUIN DD *
SSID DEGA
WORKID RDACRJ.MIGRATE2
INCLUDE (DATA SQL AMS )
SORTDEVT SYSDA
SMSINCLUDE ( )
NOREBUILD
STANDALONESTATS ALLSTATSUPD TABLEALL
NOHISTORY
BMCCOPY
IBMLOAD   NOUTILCOPY
DYNUNLD
DYNCOPY
NOPARTCOPY
COPYDDN (COPY01 )
REGISTER (COPY01 )
NOREGENIDENTITY
SINGLEPHASE
NOVVALPROP
IBMUNLOAD
IBMRERG BMCCHECK
SYNCPONIT 10
NODBRMLIB
ORDERBY
UNLOADTABLE
UNLOADEMPTY
NOSTOPCOMMIT
TABLEACCESS
```
Figure 47  Batch Analysis JCL (part 3 of 7)
Figure 47  Batch Analysis JCL (part 4 of 7)

```
PCPY1_PRIQTY=10
PCPY1_RETLPD=
PCPY1_SECQTY=2
PCPY1_STACK=N
PCPY1_STORCLASS=
PCPY1_STORCLASS_ALT=
PCPY1_SUPPRESS_SUFF=N
PCPY1_THRESH=0
PCPY1_UNIT_ALT=
PCPY1_UNIT=SYSDA
PCPY2_DATACLASS_ALT=
PCPY2_DATACLASS=
PCPY2_EXPDT=
PCPY2_MGMTCLASS_ALT=
PCPY2_MGMTCLASS=
PCPY2_PREFIX=&PREFIX..&OBNOD
PCPY2_PRIQTY=10
PCPY2_RETLPD=
PCPY2_SECQTY=2
PCPY2_STACK=N
PCPY2_STORCLASS=
PCPY2_STORCLASS_ALT=
PCPY2_SUPPRESS_SUFF=N
PCPY2_THRESH=0
PCPY2_UNIT_ALT=
PCPY2_UNIT=SYSDA
RCPY1_DATACLASS_ALT=
RCPY1_DATACLASS=
RCPY1_EXPDT=
RCPY1_MGMTCLASS_ALT=
RCPY1_MGMTCLASS=
RCPY1_PREFIX=&PREFIX..&OBNOD
RCPY1_PRIQTY=10
RCPY1_RETLPD=
RCPY1_SECQTY=2
RCPY1_STACK=N
RCPY1_STORCLASS=
RCPY1_STORCLASS_ALT=
RCPY1_SUPPRESS_SUFF=N
RCPY1_THRESH=0
RCPY1_UNIT_ALT=
RCPY1_UNIT=SYSDA
RCPY2_DATACLASS_ALT=
RCPY2_DATACLASS=
RCPY2_EXPDT=
RCPY2_MGMTCLASS_ALT=
RCPY2_MGMTCLASS=
RCPY2_PREFIX=&PREFIX..&OBNOD
RCPY2_PRIQTY=10
RCPY2_RETLPD=
RCPY2_SECQTY=2
```
Figure 47  Batch Analysis JCL (part 6 of 7)

```
UNLD1_SECQTY=2
UNLD1_STACK=N
UNLD1_STORCLASS_ALT=
UNLD1_STORCLASS=
UNLD1_THRESH=0
UNLD1_UNIT_ALT=
UNLD1_UNIT=SYSDA
UNLD2_DATACLASS_ALT=
UNLD2_DATACLASS=
UNLD2_EXPDT=
UNLD2_MGMTCLASS_ALT=
UNLD2_MGMTCLASS=
UNLD2_PREFIX=&PREFIX..&WKID
UNLD2_PRIQTY =10
UNLD2_RETPD=
UNLD2_SECQTY =2
UNLD2_STACK=N
UNLD2_STORCLASS_ALT=
UNLD2_STORCLASS=
UNLD2_THRESH=0
UNLD2_UNIT=SYSDA
UNLD3_DATACLASS=
UNLD3_EXPDT=
UNLD3_MGMTCLASS=
UNLD3_PREFIX=&PREFIX..&OBNOD
UNLD3_RETPD=
UNLD3_STORCLASS=
UNLD3_UNIT=SYSDA
UNLD4_PREFIX=&PREFIX..&OBNOD
UNLD4_STORCLASS=
UNLD4_UNIT=SYSDA
UNLD4_MGMTCLASS=
UNLD4_DATACLASS=
UNLD_FREF_DATACLASS=
UNLD_FREF_MGMTCLASS=
UNLD_FREF_PREFIX=&PREFIX..&MSSID..&WORKID8
UNLD_FREF_STORCLASS=
UNLD_FREF_UNIT=SYSDA
UNLD_FREF_PRIQTY=10
UNLD_FREF_DIRBLOCK=250
UNLD_FREF_SECQTY=2
USER_VAR1_CHAR=
USER_VAR2_CHAR=
USER_VAR3_CHAR=
USER_VAR4_CHAR=
USER_VAR5_CHAR=
// WORKL001 DD DISP=SHR,
//   DSN=RDACRJ.V10.WLBASE01(MIGRATE2)
偏                            
/* END OF JOB
```
G Press END to return to the Analysis JCL Processing Interface panel, as shown in Figure 48.

Figure 48 Analysis JCL Processing Interface panel

H On the Analysis JCL Processing Interface panel, press Enter to submit the JCL.

I After you submit the job, press Enter to return to the Analysis JCL Processing Interface panel.

The results of Analysis are a diagnostic output file (ALUPRINT) and a worklist.

J To review the results of the analysis of the work ID (ALUPRINT), you can split your ISPF session or you can use your normal method to review SYSOUT. The last message in the diagnostic output (Figure 49) indicates that the worklist completed successfully.
Figure 49  Analysis diagnostic output

Figure 50 shows the worklist that was written to the data set that you specified on the Analysis JCL Processing Interface panel.

Figure 50  CHANGE MANAGER migrate-type worklist (part 1 of 9)
Figure 50  CHANGE MANAGER migrate-type worklist (part 2 of 9)

* ALUIN PARMS:                           *
*   SSID DEGA                             *
*   WORKID RDACRJ.MIGRATE2                *
*   INCLUDE (DATA SQL AMS )               *
*   SORTDEVT SYSDA                        *
*   SMSINCLUDE ( )                       *
*   BMCREBUILD                           *
*   STANDALONESTATS ALLSTATSUPD TABLEALL  *
*   NOHISTORY                            *
*   BMCCOPY                              *
*   IBMLOAD     NOUTILCOPY                *
*   DYNUNLD                               *
*   DYNCOPY                               *
*   NOPARTCOPY                            *
*   COPYDDN (COPY01 )                     *
*   REGISTER (COPY01 )                    *
*   NOREGENIDENTITY                       *
*   SINGLEPHASE                           *
*   NOVVALPROP                            *
*   IBMUNLOAD                             *
*   IBMREORG BMCCHECK                     *
*   SYNCPOINT 10                          *
*   NODBRMLIB                             *
*   ORDERBY                               *
*   UNLOADTABLE                           *
*   UNLOADEMPTY                           *
*   NOSTOPCOMMIT                          *
*   TABLEACCESS                           *
*   NOPARALLEL                            *
*   REORGONLINE                           *
*   REORGALL                              *
*   REORREF                               *
**********************************************************************
-TIME 000050 '2011-02-15-08.23.05.465362' 57828126
-SSID 000100 DEGA                                                        30640286
-JCLP 000150 MIGR MSSID DEGA                                            46044303
-WKID 000200 RDACRJ.MIGRATE2                                           13270862
**********************************************************************
*** ORIGINATOR OF WORKLIST WAS WORKID RDACRJ.MIGRATE2 ON SSID DEGA ***
-MIGR 000250 PHASE-1   START OF SECTION TO BE RUN ON SENDING SYSTEM     38212756
-SYNC 000300 START OF EXECUTION SYNCPOINT                                86516832
**********************************************************************
-UNLI 000350
  TEMPLATE SYRC
    DSN 'RDACRJ.MIGRATE2.SR000001.P&PART.'
    UNIT SYSDA

    UNLOAD FROM TABLE J1BASIC.T_T01AICP
      HEADER NONE
    (}
Figure 50  CHANGE MANAGER migrate-type worklist (part 3 of 9)

```sql
COLC_1
 , COLSI_2
 , COLIN_3
 , COLSI_4
 , COLDC_5
 , COLDC_6
 , COLDC_7
 , COLVC_8
 , COLFI_9
 , COLF2_10
 , COLVC_11
 , COLDT_12
 , COLTS_13
   TIMESTAMP EXTERNAL(32)
 , COLC_14
)
UNLDDN SYRC
EBCDIC CCSID(37,0,0)
NOPAD MAXERR 1
IMPLIED_TZ -06:00

-SYNC 000400 END OF UNLOAD TABLES SECTION

-SETS 000450 SET CURRENT SQLID = 'RDACRJ'
-SET 000500 CREATE DATABASE J2BASIC
  STOGROUP SYSDEFLT
  BUFFERPOOL BP8K0

-Sync 000550 SET CURRENT SQLID = USER
-SYNC 000600 END OF DATABASE SECTION

-SETS 000650 SET CURRENT SQLID = 'RDACRJ'
-SET 000700 CREATE TABLESPACE T01AICP IN J2BASIC
  NUMPARTS 4
   PART 1
     USING STOGROUP SYSDEFLT
     PRIQTY 440
     SECQTY 220
     PCTFREE 0
   ,PART 2
     USING STOGROUP SYSDEFLT
     PRIQTY 440
     SECQTY 220
     PCTFREE 0
   ,PART 3
     USING STOGROUP SYSDEFLT
     PRIQTY 440
     SECQTY 220
     PCTFREE 0
   ,PART 4
     USING STOGROUP SYSDEFLT

```
Figure 50  CHANGE MANAGER migrate-type worklist (part 4 of 9)

```
PRIQTY   440
SECQTY   220
PCTFREE  0
)
BUFFERPOOL BPO
LOCKSIZE  ANY
LOCKMAX   10
CLOSE     NO
SEGSIZE   0

-SETS 000750 SET CURRENT SQLID = USER
-SYNC 000800 END OF CREATE TABLESPACE SECTION

***********************************************************************

-SET  000850 CREATE TABLE RDACRJ.T_T01AICP
(
  COLC_1
CHAR(40) NOT NULL WITH DEFAULT
,  COLSI_2
SMALLINT NOT NULL WITH DEFAULT
,  COLIN_3
INTEGER NOT NULL WITH DEFAULT
,  COLSI_4
SMALLINT NOT NULL WITH DEFAULT
,  COLDC_5
DECIMAL(6,2) NOT NULL WITH DEFAULT
,  COLDC_6
DECIMAL(9,0) NOT NULL WITH DEFAULT
,  COLDC_7
DECIMAL(5,2) NOT NULL WITH DEFAULT
,  COLVC_8
VARCHAR(15) NOT NULL WITH DEFAULT
,  COLF1_9
REAL NOT NULL WITH DEFAULT
,  COLF2_10
FLOAT NOT NULL WITH DEFAULT
,  COLVC_11
VARCHAR(15) NOT NULL WITH DEFAULT
,  COLDT_12
DATE NOT NULL WITH DEFAULT
,  COLTS_13
TIMESTAMP NOT NULL WITH DEFAULT
,  COLC_14
CHAR(100) NOT NULL WITH DEFAULT
)
CCSID EBCDIC
IN J2BASIC.T01AICP

-SET  000900 COMMENT ON TABLE RDACRJ.T_T01AICP
IS 'T01- FACTS'

-SYNC 000950 END OF TABLE SECTION
```
**Figure 50  CHANGE MANAGER migrate-type worklist (part 5 of 9)**

```sql
-SQL 001000 CREATE INDEX
    RDACRJ.I_T01AICP1
    ON RDACRJ.T_T01AICP
    ( COLSI_2
      , COLVC_8 )
      NOT PADDED
      CLUSTER
      (PART 1 VALUES (10,'C23456789012')
        USING STOGROUP SYSDEFLT
        PRIQTY 1440
        SECQTY 720
        PCTFREE 0
      ,PART 2 VALUES (13,'K23456789012')
        USING STOGROUP J1BASIC3
        PRIQTY 1440
        SECQTY 720
        PCTFREE 0
      ,PART 3 VALUES (16,'R23456789012')
        USING STOGROUP J1BASIC4
        PRIQTY 1440
        SECQTY 720
        PCTFREE 0
      ,PART 4 VALUES (MAXVALUE,MAXVALUE)
        USING STOGROUP SYSDEFLT
        PRIQTY 1440
        SECQTY 720
        PCTFREE 0
    )
    BUFFERPOOL BP1
    CLOSE YES

-SYNC 001050 END OF CREATE INDEX SECTION

-LDXT 001100
    TEMPLATE SYRC
        DSN 'RDACRJ.MIGRATE2.SR000001.P&PART.'
        UNIT SYSDA
    TEMPLATE SYDS
        DSN 'RDACRJ.MIGRATE2.SD000001.P&PART.'
        UNIT SYSDA
        SPACE(1,1) TRK
        LOAD DATA REPLACE
        ERRDDN SYSER001
        DISCARDS 1
        EBCDIC CCSID(37,0,0)
        WORKDDN SYSUT001
        ENFORCE NO
        REUSE INTO TABLE RDACRJ.T_T01AICP
```

Migrating DB2 objects

Chapter 2  Using the ALTER and CHANGE MANAGER interface  107
Migrating DB2 objects

Figure 50  CHANGE MANAGER migrate-type worklist (part 6 of 9)

<table>
<thead>
<tr>
<th>PART 1  INDDN SYRC</th>
<th>DISCARDDN SYDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td></td>
</tr>
<tr>
<td>COLC_1</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) CHAR(40),</td>
<td></td>
</tr>
<tr>
<td>COLSI_2</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) SMALLINT,</td>
<td></td>
</tr>
<tr>
<td>COLIN_3</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) INTEGER,</td>
<td></td>
</tr>
<tr>
<td>COLSI_4</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) SMALLINT,</td>
<td></td>
</tr>
<tr>
<td>COLDC_5</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td>COLDC_6</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td>COLDC_7</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td>COLVC_8</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) VARCHAR,</td>
<td></td>
</tr>
<tr>
<td>COLF1_9</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) FLOAT(21),</td>
<td></td>
</tr>
<tr>
<td>COLF2_10</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) FLOAT(53),</td>
<td></td>
</tr>
<tr>
<td>COLVC_11</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) VARCHAR,</td>
<td></td>
</tr>
<tr>
<td>COLDT_12</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DATE EXTERNAL,</td>
<td></td>
</tr>
<tr>
<td>COLTS_13</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) TIMESTAMP EXTERNAL(32),</td>
<td></td>
</tr>
<tr>
<td>COLC_14</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) CHAR(100))</td>
<td></td>
</tr>
<tr>
<td>INTO TABLE RDAQRJ.T_T01AICP</td>
<td></td>
</tr>
<tr>
<td>IGNOREFIELDS YES</td>
<td></td>
</tr>
<tr>
<td>PART 2  INDDN SYRC</td>
<td></td>
</tr>
<tr>
<td>DISCARDDN SYDS</td>
<td></td>
</tr>
<tr>
<td>(</td>
<td></td>
</tr>
<tr>
<td>COLC_1</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) CHAR(40),</td>
<td></td>
</tr>
<tr>
<td>COLSI_2</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) SMALLINT,</td>
<td></td>
</tr>
<tr>
<td>COLIN_3</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) INTEGER,</td>
<td></td>
</tr>
<tr>
<td>COLSI_4</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) SMALLINT,</td>
<td></td>
</tr>
<tr>
<td>COLDC_5</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td>COLDC_6</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td>COLDC_7</td>
<td></td>
</tr>
<tr>
<td>POSITION(*) DECIMAL,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 50   CHANGE MANAGER migrate-type worklist (part 7 of 9)
Figure 50  CHANGE MANAGER migrate-type worklist (part 8 of 9)

```
(  
  COLC_1  
  POSITION(*) CHAR(40),  
  COLSI_2  
  POSITION(*) SMALLINT,  
  COLIN_3  
  POSITION(*) INTEGER,  
  COLSI_4  
  POSITION(*) SMALLINT,  
  COLDC_5  
  POSITION(*) DECIMAL,  
  COLDC_6  
  POSITION(*) DECIMAL,  
  COLDC_7  
  POSITION(*) DECIMAL,  
  COLVC_8  
  POSITION(*) VARCHAR,  
  COLF1_9  
  POSITION(*) FLOAT(21),  
  COLF2_10  
  POSITION(*) FLOAT(53),  
  COLVC_11  
  POSITION(*) VARCHAR,  
  COLDT_12  
  POSITION(*) DATE EXTERNAL,  
  COLTS_13  
  POSITION(*) TIMESTAMP EXTERNAL(32),  
  COLC_14  
  POSITION(*) CHAR(100))
```

- SYNC 001150 LOAD TB RDAJCRJ.T_T01AICP COMPLETE  55714108
- SYNC 001200 END OF LOAD TABLES SECTION  77989868
**********************************************************************
- BMCC 001250
OUTPUT DCPYLOCP  
UNIT SYSDA  
DSNAME &UID.&DB.&TS.LP000001  
COPY TABLESPACE J2BASIC.T01AICP  
COPYDDN (DCPYLOCP) 63388934
- SYNC 001300 COPY OF MULTIPLE TABLESPACES COMPLETE  15628781
- SYNC 001350 END OF COPY SECTION  17761493
**********************************************************************
- BMCS 001400
BMCSTATS TABLESPACE J2BASIC.T01AICP  
TABLE ALL  
UPDATEDB2STATS YES  
KEYCARD Y  
REPORT NO 18183789
- SYNC 001450 STATS FOR TS J2BASIC.T01AICP COMPLETE  41779430
K Press END.

The Execution JCL Build Interface panel is displayed, as shown in Figure 51.

Figure 51 Execution JCL Build Interface panel

5 Execute the worklist.

The Execution component executes the migrate worklist in a single phase. First, the data is unloaded from the objects. Then, the objects are created, the data is loaded, and the utilities are executed.

The Execution component of the product uses the worklist as its input job stream and performs the tasks contained in the worklist. The worklist acts as a detailed procedure for implementing the changes that you defined in the specification stage.

A On the Execution JCL Build Interface panel, press Enter to accept the defaults.

The Execution Pre- and Post-Processing Interface panel is displayed, as shown in Figure 52.
On the Execution Pre- and Post-Processing Interface panel, press Enter to accept the defaults.

The Execution JCL Processing Interface panel is displayed, as shown in Figure 53.

**Figure 52** Execution Pre- and Post-Processing Interface panel

```
ACMFEXC1 -------------- Execution Pre- and Post-Processing Interface --------------
Command ===> \\
WORKID . . . . : RDACRJ.MIGRATE2 \\
Specify additional steps to be included in this job. Then press Enter. \\
Pre-Execution Compare \\
  1. Do not do a compare before execution \\
  2. Compare two previous baselines \\
  3. Compare a previous baseline to the current catalog \\
Pre-Execution Baseline \\
  1. Do not build a baseline before execution \\
  2. Build a baseline before execution \\
Post-Execution Compare \\
  1. Do not do a compare after execution \\
  2. Compare a previous baseline to the new catalog structures \\
  3. Compare the baseline built during a previous job step to the new catalog structures \\
  4. Create CDL to fall back to a previous baseline \\
Post-Execution Baseline \\
  1. Do not build a baseline after execution \\
  2. Build a baseline after execution \\
Commands: HELP PREVIOUS END
```

**Figure 53** Execution JCL Processing Interface panel

```
ACMFEXC3 -------------- Execution JCL Processing Interface --------------
Command ===> \\
WORKID . . . . : RDACRJ.MIGRATE2 \\
Specify Dataset Names. \\
Execution JCL . . . 'RDACRJ.V10.EXECJCL(MIGRATE2)'
Worklist . . . . . 'RDACRJ.V10.ANLYWL(MIGRATE2)'
Diagnostics . . . SYSOUT \\
JCL Build Options. \\
  Dataset Sizing : No Sizing (use Override Defaults to set or change) \\
Select Processing Options. Then press Enter to continue. \\
  _ Override Defaults \\
  _ Edit Worklist \\
  S Build Execution JCL \\
  S Edit Execution JCL \\
  S Submit Execution JCL \\
Commands:  BROWSE VIEWX PREVIOUS HELP END
```
On the Execution JCL Processing Interface panel, specify your data set names, specify \texttt{SYSOUT} for diagnostics, accept the defaults for your build option, and type \texttt{S} to select your processing options. Press Enter.

The Execution JCL is displayed, as shown in Figure 54. The JCL includes the \texttt{//AEXIN DD} statement (input stream), which includes keywords that are used in executing the JCL. For more information about the input stream, see the \textit{ALTER and CHANGE MANAGER for DB2 Reference Manual}.

\textbf{Figure 54 Execution JCL for a migrate-type worklist (part 1 of 4)}

\begin{verbatim}
//RDACRJE JOB (5715),'EXECUTE-MIGRATE2',
//  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//  NOTIFY=RDACRJ
/**
/* *************************************************************************/
/*  CREATED BY :  RDACRJ
/*  TIMESTAMP  :  02/15/2011.08.51.34
/*  ENVIRONMENT:  ISPF 6.1MVS     TSO
/*  RELEASE    :  V10.01.00 03/24/2011
/*  DB2 VERSION:  1000
/* *************************************************************************/
/*-*************************************************************************
/// BMC CHANGE MANAGER EXECUTION
/*-*************************************************************************
///STEP1 EXEC PGM=AEXEMAIN,
///  REGION=0M,MEMLIMIT=NOLIMIT,PARM='DMA1SEGA',
///  COND=(4,LT)
///STEPLIB DD  DSN=SYS3.DEFF.DSNEXIT,DISP=SHR
///   DD  DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
///    DD DISP=SHR,DSN=ACM.SMP1010.DBLINK
///*
///ABNLIGNR DD DUMMY
///SYSUDUMP DD SYSOUT=* 
///SYSTERM  DD SYSOUT=* 
///UTPRINT  DD SYSOUT=* 
///*************************************************************************
///  STATISTICS WORK DD STATEMENTS
///*************************************************************************
///STPRINO1 DD SYSOUT=* 
///RNPRINO1 DD SYSOUT=* 
///SYSOUT   DD SYSOUT=* 
///AEXIN    DD *
ACM 
SSID DEGA WORKID RDACRJ.MIGRATE2
DYNWORKUNIT SYSDA
DASDDOPT DS10SRGA
CATDOPT DC101DOP
STOPWAIT 3
STOPWTSECS 10
\end{verbatim}
Figure 54  Execution JCL for a migrate-type worklist (part 2 of 4)
Figure 54  Execution JCL for a migrate-type worklist (part 3 of 4)

```sql
//** ZERO VALUES WERE RETURNED FOR ONE OR MORE PARTS OF TB JIBASIC.T_J01AICP
//SYSMAP DD DSN=RDACRJ.MIGRATE2.DEAG.SYSMAP,
//   DISP=(NEW,CATLG,CATLG),
//   SPACE=(CYL,(1,1)), SPACE ESTIMATE-C/W
//   UNIT=SYSDA
//**

//** ERRDDN OUTPUT DD STATEMENTS

//** M = ESTIMATE IS THE MAXIMUM DATASET SIZE FOUND FOR THIS WORKDD TYPE
//SYSER001 DD DSN=RDACRJ.MIGRATE2.STEP1.SYSER001,
//   DISP=(NEW,CATLG,CATLG),
//   SPACE=(CYL,(1,1),RLSE), ESTIMATE-C/M
//   UNIT=SYSDA
//**

//** SORT WORK DD STATEMENTS

//** S = SORTWK SIZE IS (WORK SPACE * 2 / #SORTWKS)
//SORTWK01 DD UNIT=SYSDA,
//   SPACE=(CYL,(4,1)), ESTIMATE-C/S
//   DISP=(NEW,DELETE)
//SORTWK02 DD UNIT=SYSDA,
//   SPACE=(CYL,(4,1)), ESTIMATE-C/S
//   DISP=(NEW,DELETE)
//SORTWK03 DD UNIT=SYSDA,
//   SPACE=(CYL,(4,1)), ESTIMATE-C/S
//   DISP=(NEW,DELETE)
//SORTWK04 DD UNIT=SYSDA,
//   SPACE=(CYL,(4,1)), ESTIMATE-C/S
//   DISP=(NEW,DELETE)
//SORTWK05 DD UNIT=SYSDA,
//   SPACE=(CYL,(4,1)), ESTIMATE-C/S
//   DISP=(NEW,DELETE)
//**

//** END OF JOBSTEP

//**

//** DATASET CLEANUP STEP WILL EXECUTE IF PREVIOUS CONDITION CODE IS 4 OR LESS.

//CLEANUP EXEC PGM=IEFBR14,
//   COND=(4,LT)
//SYSUTO01 DD DSN=RDACRJ.MIGRATE2.STEP1.SYSUTO01,
//   DISP=(MOD,DELETE,DELETE),
//   SPACE=(TRK,(1,1)),
//   UNIT=SYSDA
//SORTOUT DD DSN=RDACRJ.MIGRATE2.STEP1.SORTOUT,
//   DISP=(MOD,DELETE,DELETE),
//   SPACE=(TRK,(1,1)),
//   UNIT=SYSDA
//SYSMAP DD DSN=RDACRJ.MIGRATE2.DEAG.SYSMAP,
```
Migrating DB2 objects

After you review the JCL, press END to return to the Execution JCL Processing Interface panel.

From the Execution JCL Processing Interface panel, press Enter to submit the JCL and migrate your objects.

The result of execution is a worklist execution log (AEXPRINT).

To review the results of execution in the worklist execution log (AEXPRINT), you can split your ISPF session or you can use your normal method to review SYSOUT. Figure 55 shows the worklist execution log.

The last message in the worklist indicates that the work ID status is complete and that the worklist completed successfully. The worklist migrated the objects that you specified.

Figure 54 Execution JCL for a migrate-type worklist (part 4 of 4)

```
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
//SYSER001 DD DSN=RDACRJ.MIGRATE2.STEP1.SYSER001,
//         DISP=(MOD,DELETE,DELETE),
//         SPACE=(TRK,(1,1)),
//         UNIT=SYSDA
/*-----------------------------*/
/* END OF DATASET CLEANUP STEP */
/*-----------------------------*/
/* END OF JOB
/*-----------------------------*/
```

Figure 55 Execution results for a migrate-type worklist (part 1 of 15)
Figure 55  Execution results for a migrate-type worklist (part 2 of 15)
Figure 55  Execution results for a migrate-type worklist (part 3 of 15)
Figure 55  Execution results for a migrate-type worklist (part 5 of 15)

```sql
120 ALTER and CHANGE MANAGER for DB2 Getting Started Guide

(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.
2011-02-15 09.21.19 ON SYSID DB2B                      WORKLIST EXECUTION LOG PAGE   7
------------------------------------------------------------------------------------------------------------------
LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS
------------------------------------------------------------------------------------------------------------------

120 ALTER and CHANGE MANAGER for DB2 Getting Started Guide

(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.
2011-02-15 09.21.19 ON SYSID DB2B                      WORKLIST EXECUTION LOG PAGE   7
------------------------------------------------------------------------------------------------------------------
LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS
------------------------------------------------------------------------------------------------------------------
```
Figure 55  Execution results for a migrate-type worklist (part 6 of 15)
Migrating DB2 objects

Figure 55  Execution results for a migrate-type worklist (part 7 of 15)

```sql
: REUSE
: INTO TABLE RDACRJ.T_TOIAICP
: IGNOREFIELDS YES
: PART 1  INDDN SYRC
: DISCARDDN SYDS
:
: ( COLC_1
:   POSITION(*) CHAR(40),
: COLSI_2
:   POSITION(*) SMALLINT,

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-15 09.21.22 ON SYSID DB2B WORKLIST EXECUTION LOG PAGE 9

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

: COLIN_3
:   POSITION(*) INTEGER,
: COLSI_4
:   POSITION(*) SMALLINT,
: COLDC_5
:   POSITION(*) DECIMAL,
: COLDC_6
:   POSITION(*) DECIMAL,
: COLDC_7
:   POSITION(*) DECIMAL,
: COLVC_8
:   POSITION(*) VARCHAR,
: COLF1_9
:   POSITION(*) FLOAT(21),
: COLF2_10
:   POSITION(*) FLOAT(53),
: COLVC_11
:   POSITION(*) VARCHAR,
: COLDT_12
:   POSITION(*) DATE EXTERNAL,
: COLTS_13
:   POSITION(*) TIMESTAMP EXTERNAL(32),
: COLC_14
:   POSITION(*) CHAR(100))

( C )
: INTO TABLE RDACRJ.T_TOIAICP
: IGNOREFIELDS YES
: PART 2  INDDN SYRC
: DISCARDDN SYDS
:
: ( COLC_1
:   POSITION(*) CHAR(40),
: COLSI_2
:   POSITION(*) SMALLINT,
: COLIN_3
:   POSITION(*) INTEGER,
: COLSI_4
:   POSITION(*) SMALLINT,
: COLDC_5
:   POSITION(*) DECIMAL,
: COLDC_6
:   POSITION(*) DECIMAL,
: COLDC_7
:   POSITION(*) DECIMAL,
: COLVC_8
:   POSITION(*) VARCHAR,
: COLF1_9
:   POSITION(*) FLOAT(21),
: COLF2_10
:   POSITION(*) FLOAT(53),
```

122  ALTER and CHANGE MANAGER for DB2 Getting Started Guide
Figure 55  Execution results for a migrate-type worklist (part 8 of 15)

```sql
: COLVC_11
  *** BMC EXECUTION FOR DB2 ***
  *** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.
2011-02-15 09.21.23 ON SYSID DB2B WORKLIST EXECUTION LOG PAGE 10
------------------------------------------------------------------------------------------------------------------
LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS
------------------------------------------------------------------------------------------------------------------
:    POSITION(*) VARCHAR,
:  COLDT_12
:    POSITION(*) DATE EXTERNAL,
:  COLTS_13
:    POSITION(*) TIMESTAMP EXTERNAL(32),
:  COLC_14
:    POSITION(*) CHAR(100))
:             INTO TABLE RDACRJ.T_T01AICP
:             IGNOREFIELDS YES
:             PART 3  INDDN SYRC
:             DISCARDDN SYDS
:
: COLC_1
:    POSITION(*) CHAR(40),
:  COLSI_2
:    POSITION(*) SMALLINT,
:  COLIN_3
:    POSITION(*) INTEGER,
:  COLSI_4
:    POSITION(*) SMALLINT,
:  COLDC_5
:    POSITION(*) DECIMAL,
:  COLDC_6
:    POSITION(*) DECIMAL,
:  COLDC_7
:    POSITION(*) DECIMAL,
:  COLVC_8
:    POSITION(*) VARCHAR,
:  COLF1_9
:    POSITION(*) FLOAT(21),
:  COLF2_10
:    POSITION(*) FLOAT(53),
:  COLVC_11
:    POSITION(*) VARCHAR,
:  COLDT_12
:    POSITION(*) DATE EXTERNAL,
:  COLTS_13
:    POSITION(*) TIMESTAMP EXTERNAL(32),
:  COLC_14
:    POSITION(*) CHAR(100))
:             INTO TABLE RDACRJ.T_T01AICP
:             IGNOREFIELDS YES
:             PART 4  INDDN SYRC
:             DISCARDDN SYDS
:
: COLC_1
:    POSITION(*) CHAR(40),
:  COLSI_2
:    POSITION(*) SMALLINT,
:  COLIN_3
```

---

Figure 55  Execution results for a migrate-type worklist (part 8 of 15)
Figure 55  Execution results for a migrate-type worklist (part 9 of 15)

```sql
: POSITION(*) INTEGER,
: COLSI_4
:   POSITION(*) SMALLINT,
: COLDC_5
:   POSITION(*) DECIMAL,
: COLDC_6
:   POSITION(*) DECIMAL,
: COLDC_7
:   POSITION(*) DECIMAL,
: COLVC_8
:   POSITION(*) VARCHAR,
: COLF1_9
:   POSITION(*) FLOAT(21),
: COLF2_10
:   POSITION(*) FLOAT(53),
: COLVC_11
:   POSITION(*) VARCHAR,
: COLDT_12
:   POSITION() DATE EXTERNAL,
: COLTS_13
:   POSITION() TIMESTAMP EXTERNAL(32),
: COLC_14
:   POSITION() CHAR(100))
```

21200738

BMC16815I  PARAMETERS: DEGA,MIGRATE2
BMC16822I  CONTROL PASSED TO 'DSNUTILB'
BMC16970I  CURRENT SQLID REESTABLISHED VIA: SET CURRENT SQLID = USER
BMC16898I  DEBUG: DSNUTILB 0

DSNU000I  046 09:21:23.09 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = MIGRATE2
DSNU1044I  046 09:21:23.13 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I  046 09:21:23.21 DSNUGUTC - TEMPLATE SYRC DSN 'RDACRJ.MIGRATE2.SR000001.P&PART.' UNIT SYSDA
DSNU1035I  046 09:21:23.21 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I  046 09:21:23.21 DSNUGUTC - TEMPLATE SYDS DSN 'RDACRJ.MIGRATE2.SD000001.P&PART.' UNIT SYSDA SPACE(1, 1)
DSNU050I  046 09:21:23.22 DSNUGUTC - LOAD DATA REPLACE ERRDDN SYSER001 DISCARDS 1 EBCDIC CCSID(37, 0, 0) WORKDDN SYSUT001 ENFORCE NO REUSE
DSNU650I  *DEGA 046 09:21:23.24 DSNURWI - INTO TABLE RDACRJ.T_T01AICP IGNOREFIELDS YES PART 1 INDDN SYRC DISCARDDN SYDS
DSNU650I  *DEGA 046 09:21:23.28 DSNURWI - (COLC_1 POSITION(*) CHAR(40),
DSNU650I  *DEGA 046 09:21:23.29 DSNURWI - COLSI_2 POSITION(*) SMALLINT,
DSNU650I  *DEGA 046 09:21:23.29 DSNURWI - COLIN_3 POSITION(*) INTEGER,
DSNU650I  *DEGA 046 09:21:23.29 DSNURWI - COLSI_4 POSITION(*) SMALLINT,
DSNU650I  *DEGA 046 09:21:23.29 DSNURWI - COLDC_5 POSITION(*) DECIMAL,
DSNU650I  *DEGA 046 09:21:23.29 DSNURWI - COLDC_6 POSITION(*) DECIMAL,

**** B M C   E X E C U T I O N   F O R   D B 2 ****
**** V E R S I O N:  V10.01.00 ****
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-15 09.21.27 ON SYSID DB2B WORKLIST EXECUTION LOG PAGE 12

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

...
Figure 55  Execution results for a migrate-type worklist (part 10 of 15)

| 046 09:21:23.30 DSNURWI | COLSI_4 POSITION(*) SMALLINT, |
| 046 09:21:23.30 DSNURWI | COLDC_6 POSITION(*) DECIMAL, |
| 046 09:21:23.30 DSNURWI | COLDC_7 POSITION(*) DECIMAL, |
| 046 09:21:23.30 DSNURWI | COLF1_9 POSITION(*) FLOAT(21), |
| 046 09:21:23.30 DSNURWI | COLF2_10 POSITION(*) FLOAT(53), |
| 046 09:21:23.30 DSNURWI | COLVC_11 POSITION(*) VARCHAR, |
| 046 09:21:23.30 DSNURWI | COLVC_12 POSITION(*) DATE EXTERNAL, |
| 046 09:21:23.30 DSNURWI | COLTS_13 POSITION(*) TIMESTAMP EXTERNAL(32), |
| 046 09:21:23.30 DSNURWI | COLC_14 POSITION(*) CHAR(100), |
| 046 09:21:23.30 DSNURWI | INTO TABLE RDACRJ.T_T01AICP IGNOREFIELDS YES PART 3 INDDN SYRC DISCARDDN SYDS |
| 046 09:21:23.30 DSNURWI | COLC_1 POSITION(*) CHAR(40), |
| 046 09:21:23.30 DSNURWI | COLSI_2 POSITION(*) SMALLINT, |
| 046 09:21:23.30 DSNURWI | COLIN_3 POSITION(*) INTEGER, |
| 046 09:21:23.30 DSNURWI | COLSI_4 POSITION(*) SMALLINT, |
| 046 09:21:23.30 DSNURWI | COLDC_5 POSITION(*) DECIMAL, |
| 046 09:21:23.30 DSNURWI | COLDC_6 POSITION(*) DECIMAL, |
| 046 09:21:23.30 DSNURWI | COLDC_7 POSITION(*) DECIMAL, |
| 046 09:21:23.30 DSNURWI | COLF1_9 POSITION(*) FLOAT(21), |
| 046 09:21:23.30 DSNURWI | COLF2_10 POSITION(*) FLOAT(53), |
| 046 09:21:23.30 DSNURWI | COLVC_11 POSITION(*) VARCHAR, |
| 046 09:21:23.30 DSNURWI | COLVC_12 POSITION(*) DATE EXTERNAL, |
| 046 09:21:23.30 DSNURWI | COLTS_13 POSITION(*) TIMESTAMP EXTERNAL(32), |
| 046 09:21:23.30 DSNURWI | COLC_14 POSITION(*) CHAR(100), |
| 046 09:21:23.30 DSNURWI | INTO TABLE RDACRJ.T_T01AICP IGNOREFIELDS YES PART 4 INDDN SYRC DISCARDDN SYDS |

2011-02-15 09.21.27 ON SYSID DB28

WORKLIST EXECUTION LOG  PAGE 13

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

**** B M C E X E C U T I O N F O R   D B 2 ****
**** V E R S I O N: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.
Figure 55  Execution results for a migrate-type worklist (part 11 of 15)

**DDNAME=SYS00012**

**DSN=RDACRJ.MIGRATE2.SD000001.P00003**

**DSNU1038I  046 09:21:23.93 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=SYRC**

**DDNAME=SYS00013**

**DSN=RDACRJ.MIGRATE2.SR000001.P00004**

**DSNU1038I  046 09:21:24.07 DSNUGDYN - DATASET ALLOCATED.  TEMPLATE=SYDS**

**DDNAME=SYS00014**

**DSN=RDACRJ.MIGRATE2.SD000001.P00004**

**DSNU350I  *DEGA 046 09:21:25.52 DSNURST - EXISTING RECORDS DELETED FROM TABLESPACE**

**DSNU364I    046 09:21:25.80 DSNURPPL - PARTITIONS WILL BE LOADED IN PARALLEL, NUMBER OF TASKS = 4**

**DSNU1148I   046 09:21:26.17 DSNURPPLD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=1 FOR PART 1**

**DSNU1148I   046 09:21:26.17 DSNURPPLD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=0 FOR PART 3**

**DSNU1148I   046 09:21:26.17 DSNURPPLD - (RE)LOAD PHASE STATISTICS - NUMBER OF INPUT RECORDS PROCESSED=0 FOR PART 2**

**DSNU303I  *DEGA 046 09:21:26.43 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=1 FOR TABLE**

**RDACRJ.T_T01AICP  PART=1**

**DSNU303I  *DEGA 046 09:21:26.43 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=0 FOR TABLE**

**RDACRJ.T_T01AICP  PART=2**

**DSNU303I  *DEGA 046 09:21:26.43 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=1 FOR TABLE**

**RDACRJ.T_T01AICP  PART=3**

**DSNU303I  *DEGA 046 09:21:26.43 DSNURWT - (RE)LOAD PHASE STATISTICS - NUMBER OF RECORDS=0 FOR TABLE**

**RDACRJ.T_T01AICP  PART=4**

**DSNU147I   046 09:21:26.46 DSNURWT - (RE)LOAD PHASE STATISTICS - TOTAL NUMBER OF RECORDS LOADED=2 FOR TABLESPACE**

**J2BASIC.T01AICP**

**DSNU348I  *DEGA 046 09:21:26.46 DSNURLD - BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX RDACRJ.I_T01AICP1**

**PART**

**DSNU348I  *DEGA 046 09:21:26.46 DSNURLD - BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX RDACRJ.I_T01AICP1**

**PART**

**DSNU348I  *DEGA 046 09:21:26.46 DSNURLD - BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX RDACRJ.I_T01AICP1**

**PART**

**DSNU348I  *DEGA 046 09:21:26.46 DSNURLD - BUILD PHASE STATISTICS - NUMBER OF KEYS=1 FOR INDEX RDACRJ.I_T01AICP1**

**PART**

**DSNU300I    046 09:21:26.46 DSNURILD - (RE)LOAD PHASE COMPLETE, ELAPSED TIME=00:00:02**

**DSNU010I    046 09:21:27.14 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0**

**:-SYNC 001150 LOAD TB RDACRJ.T_T01AICP COMPLETE                          55714108**

**:-SYNC 001200 END OF LOAD TABLES SECTION                                 77989868**

***************************************************************************

**:-BMCC 001250**

**:    OUTPUT DCPYLOCP**

**:      UNIT SYSDA**

**:      DSNAME &UID.&DB.&TS.LP000001**

**:      COPY TABLESPACE J2BASIC.T01AICP**

**:      COPYDN (DCPYLOCP)**

**:                  63388934**

**BMC16815I  PARAMETERS: DEGA,MIGRATE2**

**BMC16822I  CONTROL PASSED TO 'ACPMAIN'**

**BMC16970I  CURRENT SQLID REESTABLISHED VIA: SET CURRENT SQLID = USER**

**BMC16898I  DEBUG: ACPMAIN 0**

**BMC30101I COPY PLUS FOR DB2 V10.1.00**

**BMC47491I COPYRIGHT BMC SOFTWARE INC. 1991-2010**

**BMC47487I COPY PLUS TECHNOLOGY IS PROTECTED BY U.S. PATENTS 7,133,884 AND 7,769,718**

**BMC47492I CONTACT BMC SUPPORT AT 1-800-537-1813 OR EMAIL TO SUPPORT@BMC.COM**

**BMC30001I UTILITY EXECUTION STARTING  02/15/2011  09:21:31 ...
Figure 55  Execution results for a migrate-type worklist (part12of15)
Figure 55  Execution results for a migrate-type worklist (part 13 of 15)

```
BMC30101I  APCOMPARE
BMC30101I  APRETAINDUP
BMC30101I  DSNT255I  *DEGA DSNTBCM2 BIND OPTIONS FOR
BMC30101I  PACKAGE = DEGA.ACPB101T.ACPSQL1.(ACP_1010)
BMC30101I  SOLERERROR NOPACKAGE
BMC30101I  CURRENTDATA NO
BMC30101I  DEGREE 1
BMC30101I  DYNAMICRULES
BMC30101I  DEFER
BMC30101I  REDPT NONE
BMC30101I  KEEP_DYNAMIC NO

**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.

2011-02-15 09.21.34 ON SYSD DB28 WORKLIST EXECUTION LOG PAGE 16

LINE PREFIX : DENOTES WORKLIST INPUT STATEMENTS

BMC30101I  IMMEDWRITE NO
BMC30101I  DBPROTOCOL DRDA
BMC30101I  OPTHINT
BMC30101I  ENCODING EBCDIC(00037)
BMC30101I  PLANMGMT OFF
BMC30101I  PLANMGMTSCOPE STATIC
BMC30101I  CONCURRENTACCESSRESOLUTION
BMC30101I  EXTENDEDINDICATOR
BMC30101I  PATH
BMC30101I  DSNT232I  *DEGA SUCCESSFUL BIND FOR
BMC30101I  PACKAGE = DEGA.ACPB101T.APCSQL1.(ACP_1010)
BMC30101I  USING STANDARD DB2 SECURITY, EXIT(12)
BMC30101I  EBIDIC CCSID = 37
BMC30101I  DB2 VERSION = 1010 SITE TYPE = LOCAL MODE = NFM
BMC30101I  RETURN CODE 4 FROM DYNAMIC BIND OUTPUT ACB
BMC30101I  BMCACP.BMCACP_GROUPS IS NOT DEFINED

BMC30101I  DSNT254I  *DEGA DSNTBCM2 BIND OPTIONS FOR
BMC30101I  PACKAGE = DEGA.ACPB101T.APCSQL1.(ACP_1010)
BMC30101I  ACTION REPLACE
BMC30101I  OWNER CSTCXN
BMC30101I  VALIDATE RUN
BMC30101I  EXPLAIN NO
BMC30101I  ISOLATION
BMC30101I  RELEASE
BMC30101I  COPY
BMC30101I  APREUSE
BMC30101I  APCOMPARE
BMC30101I  APRETAINDUP
```
Figure 55 Execution results for a migrate-type worklist (part 14 of 15)
Figure 55  Execution results for a migrate-type worklist (part 15 of 15)

```sql
BMC278768I UTILID: MIGRATE2 INITIALIZATION COMPLETE
BMC278611I STATISTICS ANALYSIS PHASE INITIATED
BMC348012I BMCSTATS statistics collection mode set to ENHANCED.
BMC278762I STATISTICS ANALYSIS PHASE COMPLETE HIGHEST RETURN CODE 0
BMC278769I JOB TOTALS: OBJECTS SELECTED FROM CATALOG : 2
BMC278770I JOB TOTALS: TS(s) SELECTED FROM CATALOG : 1
BMC278771I JOB TOTALS: TS(s) SUCCESSFULLY PROCESSED : 1
BMC278772I JOB TOTALS: TS(s) OBJECTS BYPASSED : 0
BMC278773I JOB TOTALS: TS(s) PROCESSING ERRORS : 0
BMC278774I JOB TOTALS: IX(s) SELECTED FROM CATALOG : 1
**** BMC EXECUTION FOR DB2 ****
**** VERSION: V10.01.00
(C) COPYRIGHT 1991 - 2011 BMC SOFTWARE, INC.
```

2011-02-15 09.21.39 ON SYSSID DB2B  WORKLIST EXECUTION LOG  PAGE 19

```
BMC278771I JOB TOTALS: IX(s) SUCCESSFULLY PROCESSED : 1
BMC278772I JOB TOTALS: IX(s) OBJECTS BYPASSED : 0
BMC278773I JOB TOTALS: IX(s) PROCESSING ERRORS : 0
BMC278774I JOB TOTALS: PARTS SCHEDULED : 8
BMC278775I JOB TOTALS: PARTS PROCESSED : 8
BMC278014I UTILID: MIGRATE2 RESET/TERMINATION COMPLETE
:-SYNC 001450 STATS FOR TS J2BASIC.T01AICP COMPLETE 41779430
:-SYNC 001500 END OF STATISTICS SECTION 37900482
***********************************************************************
BMC16999I  UPDATED WORKID STATUS FOR RDACRJ.MIGRATE2 IS 'C' (COMPLETE)
BMC16833I  END OF WORKLIST EXECUTION
```
Press END to return to the Migrate DB2 Objects Using Specification panel.

Press END to return to the Task List Menu panel.

Press END to return to the CHANGE MANAGER Main Menu.

This example shows that with CHANGE MANAGER you have the ability to quickly and accurately specify objects that you want to migrate to another environment, perform an in-depth analysis of the objects selected to determine their effect on your subsystem, and execute SQL statements, utilities, and other commands.

Where to go from here

Now that you have an idea of how you can use ALTER or CHANGE MANAGER in your environment, see Chapter 3, “Setting up ALTER and CHANGE MANAGER” to customize the products for your environment.
Where to go from here
Chapter 3 Setting up ALTER and CHANGE MANAGER

This chapter presents the following topics:

Overview ................................................................. 134
Using options to control your environment ............................. 135
  Using the installation options ...................................... 135
  Using the user options .............................................. 136
  Using the product options .......................................... 137
  Using local keywords to override options ............................ 139
Creating an additional installation options module .................... 139
Putting it all together .................................................. 140
Allocating data sets ..................................................... 141
Naming objects .......................................................... 141
  Using wildcard patterns .............................................. 142
  Using delimited identifiers ......................................... 144
Using nonprintable or nonviewable characters ......................... 145
Controlling access to components ..................................... 145
Switching authorization .................................................. 149
  Using the Execution security exit function codes ................. 149
Testing the Execution security exit .................................... 152
Restricting access to the worklist parallelism feature ................ 152
  Applying a zap ....................................................... 153
  Specifying a general resource profile .............................. 153
Setting the MEMLIMIT system parameter ............................. 154
Modifying the ISPF skeletons and variables ............................ 154
Sizing the data sets ..................................................... 155
Maintaining the product tables ......................................... 156
  Maintaining the size of tables ...................................... 159
  Securing tables ...................................................... 160
Setting options .......................................................... 163
  Setting the attributes for the panel display ....................... 164
  Setting the options for data set allocation ....................... 166
  Setting the processing options for Analysis ...................... 167
  Setting the processing options for Import ....................... 169
Overview

This chapter describes how to set options for how ALTER and CHANGE MANAGER operate, restrict access to the components of the products, and maintain the tables used by the products.

Before you set up the ALTER or CHANGE MANAGER product for the ISPF interface, ensure that you have completed the tasks necessary to install the products. For information, see the installation guide.
Using options to control your environment

For your production requirements, you should evaluate the option settings for ALTER and CHANGE MANAGER. For most situations, the installation default values are sufficient. However, you might want to customize the operating environment and panels.

ALTER and CHANGE MANAGER use default, user, and product options to define the operating environment and to specify how the products’ components work. The options also contain default values for data set names and allocations, job control language (JCL) generation information, and component plan names.

These options provide you with the ability to

- tailor the interface
- set up defaults for generating job statements
- specify defaults for parameters, names, and prefixes for allocated data sets
- specify processing options for the Analysis, JCL Generation, Execution, Compare, Import, Baseline, and CM/PILOT components

Typically, the person who installs ALTER or CHANGE MANAGER sets default values for user options. For information about establishing installation option values at installation, see the installation guide.

---

**TIP**

For a list of tasks that you can perform to set your options, see Table 19 on page 163.

---

Using the installation options

The default operating environment is controlled by a number of option values that are defined in the installation options module. The components of ALTER and CHANGE MANAGER use the global values that are stored in the installation options module to determine how to process information. The Installation System generates the installation options module when you install ALTER or CHANGE MANAGER. The module contains an assembly-language program with an options macro.

You can customize ALTER or CHANGE MANAGER for all users by editing the default values in the installation options module. The default name of the ALTER module is ALUDOPD1 and the CHANGE MANAGER module is ACMDOPD1. The source of the installation option modules is located in the HLQ.UDBCNTL data set. HLQ identifies the high-level qualifier that you specify when you install the products.
Using the user options

The first time that you run ALTER or CHANGE MANAGER, the product copies the values from the installation options module into your ISPF profile. The values in this profile are referred to as your user options. The product uses these user options to generate JCL and to generate keywords for an input stream for each user’s subsystem. The products use the following input streams:

- AEXIN, which is used by the Execution component
- AEXPIN, which is used by the Execution component for the worklist parallelism feature in the Database Administration solution
- AJXIN and AJXPOFIN, which are used by the Batch Execution JCL Generation component

**NOTE**
AJXPOFIN is also used by Batch Analysis JCL Generation.

- ALUIN, which is used by the Analysis, Baseline, Baseline Report, Compare, CM/PILOT, and Import components

**Storing values in the ISPF profile**

The ISPF profile resides in the data set member prdxPROF, where prdx is the value of the application ID in the BMCDB2 CLIST. To define or modify the values in this profile, you can use the Options panels of ALTER or CHANGE MANAGER, JCL Generation, or CM/PILOT. You can use literal characters or symbolic variables to specify the values on the Options panels. (For more information about symbolic variables, see the ALTER or CHANGE MANAGER for DB2 Reference Manual.)

To save your user options in your profile, you must exit the product. If your ISPF session abnormally ends (abends), the user options that you modified are not saved.

**NOTE**
ALTER and CHANGE MANAGER use plan names directly. If you need to specify different plan name values for each DB2 subsystem, you must have multiple installation option modules.
Chapter 3 Setting up ALTER and CHANGE MANAGER

Using the product options

Refreshing the user options

You can refresh the values in your user options by editing and reassembling the installation options module. To refresh an option value in all existing ISPF profiles, type a comma and an R after the option value and then enclose them in parentheses, as shown in the following example:

BMCCOPY=(N,R),

NOTE
Do not remove the comma after the right parenthesis or the continuation character (*) in column 72, except for the last option value.

The next time that you run the product, the new global value replaces the old local value in the user options. You can modify the local value through the Options panels. If you need to change the installation options after installation, you must reassemble the installation options module.

For more information about refreshing user options, see the installation guide.

Using the product options

The POFDS keyword (in the installation options module) specifies an 80-character sequential file. This product options file (POF) is built during product installation and contains parameters and values for the JCL Generation, ALTER, and CHANGE MANAGER options. The file is located in the HLQ.UDBCNTL data set. The POF does not require assembly and linkage and does not need to reside in an APF-authorized data set.

The Batch Execution JCL Generation component uses the AJXIN keyword, POFDS, to specify the POF data set. Batch Execution JCL Generation can use an input stream with the ddname of AJXPOFIN to override keywords in the data set that the POFDS keyword specifies. For more information, see the ALTER and CHANGE MANAGER for DB2 User Guide.

When you install the products, only one POF is created. This initial POF is initialized and populated with the default ISPF variables and values from the installation panels. This POF is shared among several products, if those products are installed at the same time.
In addition, the installation system will use the same application ID (or profile) for the products in the BMCDB2 CLIST. This single application ID enables the JCL Generation options to be shared with other products, such as the BMC CATALOG MANAGER product. Thus, when you specify an option for generating JCL in one product, your selection applies to all of the products. Although BMC recommends that you use a single application ID, you can choose individual product application IDs on the BMCDB2PR panel.

JCL Generation also handles user POFs, which are POFs that can be written from the ISPF variables that are set in the Front End component of ALTER or CHANGE MANAGER or edited. You can use a user POF to reset all of the options that you will use in the current session to create JCL. You can also use the user POFs to set options for different sets of applications, particularly if the applications have different naming standards.

**NOTE**

If a POF keyword in your user POF uses a library from an earlier version of the product, update the keyword to use a library for the most recently installed version of the product. For example, assume that the value of the BMC_COPY_LOAD keyword is BMC9300.ACM.D91.LOAD and then you installed version 10.1.00 of the product. Update the value of the keyword to a version 10.1.00 LINK library (BMC1010.ACM.D10.DBLINK).

JCL Generation uses the variables in the ISPF profile when generating JCL. When you start ALTER or CHANGE MANAGER, JCL Generation determines whether to reset the variables in the ISPF profile:

- The first time that the product is invoked, all of the values in the ISPF profile are set to the values that are in the initial POF. If a POF is not specified, default values are assigned to the variables in the profile.

- If the POFDATE keyword in the initial POF is greater than the value of the POF date that is stored in the ISPF profile, the values in the POF that are marked with refresh *(R)* are used to reset the ISPF variables.

- If you specify a new initial POF in the POFDS installation option, the values in the POF that are marked with refresh *(R)* are used to reset the ISPF variables. The value of the POFDATE keyword in the new initial POF is saved in the ISPF profile.

For more information about POFs, see the installation guide.
Using local keywords to override options

You can override the user options for an ALTER or CHANGE MANAGER component by using the Override Options panels for that component. For example, you can override the product options for JCL Generation by using the Override Options panels for Execution. (You can also override the options that are related to dynamic allocation for JCL Generation in the same panels for Analysis.) The override options remain in effect only for the current ALTER or CHANGE MANAGER session.

The Front End uses the override options, as well as the user options, to generate the input streams for the other components. The local keywords in an input stream can override some of the values in the POF and ISPF profile. For example, when foreground processing is selected, the Analysis component’s ALUIN input stream is generated when you select Create Input from the Analysis JCL Processing Interface panel.

**NOTE**

If you do not use the Front End component to run other components, you must explicitly specify (in each component’s input stream in the JCL) any local values that you want to use that are different from the global installation option values that are defined at installation.

Creating an additional installation options module

When you issue the **DOPTS** command on the **Command** line of any ALTER or CHANGE MANAGER panel, the products invoke an ISPF browse session. In this browse session, the installation option keywords and values are displayed.

**NOTE**

This list of installation option keywords does not include the keywords that the keywords in the POF replace.

You can copy the keywords in the browse session to a permanent data set. In your stand-alone installation options job in the installation JCL data set, specify the name of this permanent data set as the data set name in the //ASM.SYSIN DD card.
Putting it all together

Figure 56 illustrates how ALTER and CHANGE MANAGER use the installation options, user options, product options, and override options.

**Figure 56  How the products use options**

* The only input stream that you can edit is the ALUIN input stream in foreground Analysis.
Allocating data sets

When you use the Front End to run components of ALTER or CHANGE MANAGER, you specify data sets for JCL, diagnostic output, worklist files, Change Definition Language (CDL) files, and report files. When you specify these data sets, use the guidelines in Table 3.

**NOTE**
You must predefine partitioned data sets (PDSs). If a sequential file or member of a PDS does not exist, the product dynamically creates or allocates it.

### Table 3 Guidelines for data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Type</th>
<th>Record format</th>
<th>Record length</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL</td>
<td>sequential file</td>
<td>fixed length (FB)</td>
<td>80 characters</td>
</tr>
<tr>
<td></td>
<td>member of PDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>diagnostic output</td>
<td>sequential file</td>
<td>variable length (VB)</td>
<td>137 characters</td>
</tr>
<tr>
<td>worklist</td>
<td>sequential file</td>
<td>fixed length (FB)</td>
<td>80 characters</td>
</tr>
<tr>
<td></td>
<td>member of PDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDL file</td>
<td>sequential file</td>
<td>fixed length (FB)</td>
<td>80 characters</td>
</tr>
<tr>
<td></td>
<td>member of PDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>report</td>
<td>sequential file</td>
<td>fixed length (FBA)</td>
<td>80 characters</td>
</tr>
<tr>
<td></td>
<td>member of PDS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**
The implicit owner for a baseline is the owner of the profile that was used to create the baseline.

Naming objects

You must name an ALTER or CHANGE MANAGER object when you create it. Work IDs, profiles, and baselines have a two-part name in the following format:

`owner.identifier`

The `owner` can be from one to eight characters, and the `identifier` can be from one to 18 characters. The two parts are subject to the same restrictions as DB2 identifiers: you cannot use delimited identifiers or double-byte character set (DBCS) characters. For information about delimited identifiers, see “Using delimited identifiers” on page 144. If you do not explicitly specify the owner, the owner defaults to your user ID.

**NOTE**

The implicit owner for a baseline is the owner of the profile that was used to create the baseline.
Using wildcard patterns

Wildcard patterns are used throughout the product to specify a partial match in the name of an object. Table 4 describes how wildcard patterns can be used in various components of ALTER and CHANGE MANAGER.

Table 4  Using wildcards

<table>
<thead>
<tr>
<th>Component</th>
<th>Use of wildcards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>to produce lists of objects</td>
</tr>
<tr>
<td>Scope rules</td>
<td>in object names, to include lists of objects under a single scope rule</td>
</tr>
<tr>
<td>Change rules</td>
<td>to find object names that changes are applied to and to supply old and new change rule values</td>
</tr>
</tbody>
</table>

Table 5 lists the wildcard characters that are supported for the Specification filters.

Table 5  Wildcard characters for Specification filters

<table>
<thead>
<tr>
<th>Wildcard character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>? (question mark)</td>
<td>matches any single character, if the data type of the object is fixed length or variable length</td>
</tr>
<tr>
<td>_ (underscore)</td>
<td>For example, a value of AB?D or AB_D matches ABCD or AB1D, but not ABD.</td>
</tr>
<tr>
<td>% (percent sign)</td>
<td>matches any single character or a string of zero or more characters, depending on the data type of the object</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>If the data type of the object is fixed length, an embedded wildcard matches any single character. For example, a value of %DB or <em>DB for a database name matches ADB or BDB, but not DB or XYZDB. A trailing wildcard matches a string of zero or more characters. For example, a value of %DB</em> for a database name matches ADB or ADB123, but does not match DB123.</td>
</tr>
<tr>
<td></td>
<td>If the data type of the object is variable length, an embedded or a trailing wildcard matches a string of zero or more characters. For example, a value of *TB or %TB matches TB, ATB, or ABTB, but not ABTBZ.</td>
</tr>
</tbody>
</table>
Table 6 lists the wildcard characters that are supported for scope rules.

### Table 6  Wildcard characters for scope rules

<table>
<thead>
<tr>
<th>Wildcard character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>? (question mark)</td>
<td>matches any single character</td>
</tr>
<tr>
<td>_ (underscore)</td>
<td>For example, a value of AB?D or AB_D matches ABCD, AB1D, but not ABD.</td>
</tr>
<tr>
<td>% (percent sign)</td>
<td>matches any string of zero or more characters</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>For example, a value of AB*D or AB%D matches ABCD, AB123D, or ABD, but not AB.</td>
</tr>
</tbody>
</table>

If you use change rules and if the new value contains wildcard characters, the current value must have exactly the same number of wildcard characters in the same order. If the new value does not contain wildcards, the current value can use any combination. Table 7 lists the wildcard characters that are supported for change rules.

### Table 7  Wildcard characters for change rules

<table>
<thead>
<tr>
<th>Wildcard character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>? (question mark)</td>
<td>matches any single character</td>
</tr>
<tr>
<td>_ (underscore)</td>
<td>For example, a current value of AB?D or AB_D matches ABCD, AB1D, but not ABD. You can create an underscore (_) as a literal instead of a wildcard by preceding it with a back slash ().</td>
</tr>
<tr>
<td>% (percent sign)</td>
<td>matches any string of zero or more characters</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>For example, a current value of AB*D or AB%D matches ABCD, AB123D, or ABD, but not AB.</td>
</tr>
</tbody>
</table>

Table 8 shows some examples of valid and invalid wildcard uses.

### Table 8  Examples of valid and invalid wildcard use for change rules  (part 1 of 2)

<table>
<thead>
<tr>
<th>Current value</th>
<th>New value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB*DE</td>
<td>X*ABC</td>
<td>valid; wildcards match</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Any characters found in the current value between the B and D characters are inserted between the X and A characters of the new value.</td>
</tr>
<tr>
<td>AB*DE</td>
<td>X<em>A</em></td>
<td>invalid; number of wildcards do not match</td>
</tr>
<tr>
<td>AB<em>DE</em></td>
<td>X*ABC</td>
<td>invalid; number of wildcards do not match</td>
</tr>
<tr>
<td>AB*DE</td>
<td>ABPROD</td>
<td>valid; no wildcards in new value</td>
</tr>
</tbody>
</table>
Using delimited identifiers

ALTER and CHANGE MANAGER support the use of delimited identifiers for DB2 objects, but not for BMC objects, such as work IDs, baselines, baseline profiles, and migrate profiles. The double quotation mark (") is the only escape character that is supported.

Delimited identifiers can also be used in data definition language (DDL), CDL, worklists, Data Manipulation Language (DML), and for reserved words. ALTER and CHANGE MANAGER evaluate the identifiers to determine whether delimiters are required. If delimiters are not required, they are removed.

Delimited identifiers are supported to specify owners of DB2 objects and to specify the names of certain types of DB2 objects. You can use delimited identifiers to specify DB2 object names that use long identifiers. These objects include tables, table columns, indexes, aliases, synonyms, views, view columns, triggers, and trigger columns. Names for databases and table spaces cannot be named using delimited identifiers. Owners of these DB2 objects, however, can be named using delimited identifiers. The following example illustrates the use of a delimited identifier:

```
000008 -SQL 000450 CREATE TABLE NCH.TBLARGE
000009   ("SYNONYM" CHAR(15) NOT NULL ,
```

**NOTE**

Twenty bytes are allowed for object names and ten bytes for owner names specified with delimited identifiers; two bytes are used for the escape characters. If you use delimited identifiers to specify names, embedded double quotation marks within a name might cause truncation. Two consecutive double quotation marks are required to represent one double quotation mark within a delimited identifier.

Wildcard characters that are specified within delimiters are considered wildcard characters, not literal characters. The asterisk (*) specified for a STOGROUP volume ID is supported for users of the IBM Storage Management Subsystem (SMS).

For additional information about using delimited identifiers, see the IBM documentation.

<table>
<thead>
<tr>
<th>Current value</th>
<th>New value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBTEST</td>
<td>AB*DE</td>
<td>invalid; no wildcards in original value</td>
</tr>
<tr>
<td>A*B_</td>
<td>X_Y*</td>
<td>invalid; wildcards out of order</td>
</tr>
</tbody>
</table>

Table 8 Examples of valid and invalid wildcard use for change rules (part 2 of 2)
Using nonprintable or nonviewable characters

You must specify nonprintable and nonviewable characters (such as null and control characters) in limit keys, view text, trigger text, or check constraint text in an external hexadecimal format. These characters are not indicated in literal strings. For example, the EBCDIC string ‘2 1’ does not indicate that the second character is null; the string appears to be blank. The string should be specified as X’F200F1’.

Table 9 provides the hexadecimal formats for common nonprintable or nonviewable characters.

Table 9  Nonprintable or nonviewable characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Hexadecimal format</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>X’00’</td>
</tr>
<tr>
<td>horizontal tab (‘HT’)</td>
<td>X’05’</td>
</tr>
<tr>
<td>form feed</td>
<td>X’0C’</td>
</tr>
<tr>
<td>carriage return</td>
<td>X’0D’</td>
</tr>
<tr>
<td>new line (‘NL’)</td>
<td>X’15’</td>
</tr>
<tr>
<td>line feed (‘LF’)</td>
<td>X’25’</td>
</tr>
</tbody>
</table>

Note the following items about nonviewable characters:

- If a value in a limit key field in a partitioned index contains a nonviewable character, you cannot edit that value in the Specification component.
- If null characters exist in the internal format of a value for an object, the Baseline, Import, and Compare components issue a warning message.

Controlling access to components

Maintaining security over the components is an important consideration. Executing a worklist can change the definitions of your DB2 objects as they are defined in the DB2 system catalog tables. To control access to the components, you control the authorization that is granted to plans for the components.

Plans that are included with ALTER and CHANGE MANAGER provide access to the Front End, Import, Specification, Analysis, Baseline, Compare, and Execution components. You can control whether users can access ALTER and CHANGE MANAGER by controlling the authorization granted to each of these plans.
ALTER and CHANGE MANAGER plan names have the format prdvryz, where prd is the product code (ALU indicates ALTER and ACM indicates CHANGE MANAGER), vr is the version and release, y is the access type, and z is the component.

The Execution component plan names have the format prdvrnn, where prd is the product code, vr is the version and release, and nn is a unique plan identifier.

**ALTER and CHANGE MANAGER plans**

Table 10 lists the plans that the components of ALTER and CHANGE MANAGER use.

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Component name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACMvryA</td>
<td>Analysis</td>
<td>enables users to access the Analysis component to analyze changes and generate worklists. Users cannot make changes to DB2 objects and data by creating a worklist. However, the Specification and Analysis plans provide a way for users to request and analyze changes. For this reason, administrators can usually place minimum restrictions on using Specification and Analysis.</td>
</tr>
<tr>
<td>ACMvryE</td>
<td>Environment</td>
<td>enables users to use the ENVI command to review the system environment. Because this plan is not accessed outside of the other components, administrators can usually grant PUBLIC access to this plan. This plan does not control the use of the ENV keyword with batch components.</td>
</tr>
<tr>
<td>ACMvryF</td>
<td>Front End</td>
<td>controls access to the ISPF interface. Authority to use this plan enables users to create and maintain the product’s objects and perform other Front End functions. If the Distributed Data Facility (DDF) is installed, the -BIND statement for this plan adds a reference to the remote package list.</td>
</tr>
<tr>
<td>ACMvryI</td>
<td>Import</td>
<td>enables users to import files into the Change Definition (CD) tables. This plan accesses the Import component. For ALTER, the plan enables users to import data definition language (DDL) into the CD tables. For CHANGE MANAGER, it enables users to import DDL, Change Definition Language (CDL), and worklists into the CD tables.</td>
</tr>
</tbody>
</table>
Controlling access to components

Chapter 3 Setting up ALTER and CHANGE MANAGER

147

Table 10 ALTER and CHANGE MANAGER plans (part 2 of 2)

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Component name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACMvryS</td>
<td>Specification</td>
<td>enables users to request changes to database objects and data. This plan accesses the Specification component. With access to Specification, users can also search and review groups of objects. Note: While Specification does not enable users to actually perform changes, it does enable them to specify changes and to view existing data structures. If the Distributed Data Facility (DDF) is installed, the -BIND statement for this plan adds a reference to the remote package list.</td>
</tr>
</tbody>
</table>

Table 11 CHANGE MANAGER plans

Table 11 lists additional plans that the components of CHANGE MANAGER use.

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Component name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACMvryB</td>
<td>Baseline</td>
<td>enables users to establish and delete baselines. Because baselines (especially full-recovery baselines) contain critical information that is used for database recovery, administrators should allow only informed users access to this plan.</td>
</tr>
<tr>
<td>ACMvryC</td>
<td>Compare</td>
<td>enables users to compare data structure definitions. Because no DB2 data structures are modified, administrators can usually grant unlimited access to this plan. If the DDF is installed, the -BIND statement for this plan adds a reference to the remote package list.</td>
</tr>
<tr>
<td>ACMvryR</td>
<td>Report</td>
<td>enables users to create baseline reports. Grant unlimited access to this plan.</td>
</tr>
<tr>
<td>ACMvryL</td>
<td>LOB DATA MOVER</td>
<td>when the Database Administration solution is installed, enables the LOB DATA MOVER program to attach to DB2. The LOB DATA MOVER program unloads and loads data that is contained in large object (LOB) columns.</td>
</tr>
</tbody>
</table>

You can restrict access to functions of the Execution component by using PLAN authorizations. Table 12 lists the plans that the Execution component uses in ALTER and CHANGE MANAGER.
## Table 12 Execution plans

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEXvrHA</td>
<td>Execution Monitor Entry (Authorization)</td>
<td>enables users to execute a worklist when EXECUTE authority is granted. You should carefully consider who receives authorization to use this plan.</td>
</tr>
</tbody>
</table>
| AEXvrHM   | Execution Monitor  | enables users to attach to DB2 with alternate authorization IDs for the -AUTH commands. This plan does not control who has authorization to execute a worklist. Because this plan does not affect who can run Execution, you can grant PUBLIC authority to this plan. The Execution plan contains some packages that use dynamic SQL. Some of these packages cause long-running SQL and might need to be added to your resource limit specification table (RLST). The packages are as follows:  
- AEXAUNLD unloads data from tables.  
- AEXSQLIO performs all worklist -SQL commands, including deletions before a -LOAD or -BMCL command that migrates data only.  
- AEXESTDL performs some of the restart logic before the -LOAD command restarts, including deleting previously loaded rows.  

By restricting authorization to run the Execution plans, you can control what change and migrate functions users can perform. For example, by granting unlimited access to Specification and Analysis while controlling access to the Execution Monitor Entry, you can allow your users to run ALTER for training purposes or use it as a system dictionary.  

The Execution Security Exit provides further control over the Execution component’s authorization switching function.
Switching authorization

The Execution component performs an authorization switching function to ensure that DB2 structures are created with the proper owner and creator. To secure this function, you can perform one of the following activities:

- Restrict the EXECUTE authority on the Execution Monitor Entry plan. This restriction will prevent unauthorized users from being able to run Execution. For the highest level of security, restrict access to the Execution Monitor Entry plan.

- Create an Execution security exit to add site-specific restrictions to authorization switching.

Execution calls the security exit at various points to allow installation security checking and option enforcement.

Typical uses of this exit include

- enforcing hash checking so that users are not allowed to modify worklists
- switching the authorization ID of the execution run user at initialization
- preventing or restricting user authorization ID switching
- preventing user modification of SQL GRANT commands
- inspecting and modifying SQL commands before they are executed

The assembler source for the security exit is distributed in the DBSAMP library (member ALUEUSX1). To enable the security exit, modify ALUEUSX1, then assemble and link the member. This module is loaded during initialization.

ALUEUSX1 communicates via Registers 0, 1, and 15. When ALUEUSX1 is called, Register 0 contains a function code. Register 1 contains the address of a user exit control block that AEXUSERX DSECT (in the sample program) describes. ALUEUSX1 returns a value in Register 15 that informs Execution how to proceed.

For more information about switching authorization, see the ALTER and CHANGE MANAGER for DB2 User Guide.

Using the Execution security exit function codes

Table 13 lists and describes the Register 0 function codes. The sample exit includes detailed documentation about register usage, DSECT mapping, and call conventions. For a list of return codes, see the Administrative Products for DB2 Messages Manual.
### Table 13  Execution security exit function codes (part 1 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Call</th>
<th>Occurs</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’01’</td>
<td>initialization</td>
<td>during Execution initialization</td>
<td>At this point, you can inspect the logon ID that is running the job, change it if necessary, or halt processing. You can also enable hash checking to prevent a user from making any modifications to a worklist (other than deleting entire commands) throughout the execution run. If you need storage for local routines, you should perform a GETMAIN at this time. Store the address in the user exit control block AEXUSERX, as described in the code sample.</td>
</tr>
<tr>
<td>X’02’</td>
<td>-AUTH command switch</td>
<td>immediately before an authorization ID switch</td>
<td>The authorization ID might be inspected or changed, or processing might be halted. If the AEXEFAUH bit of flag ASXEFLAG in the user exit DSECT is set, hash checking has failed on the current -AUTH command. If the return code from this exit is an odd number (that is, the low order 1 bit is set), the command executed under the switch is verified to ensure that the hash code is correct. This verification prevents the user from changing commands that are executed under a switch. If this bit is off (return codes 0 or 4), hash checking is done as specified for the main run. If this bit is on (return codes 1 or 5), then hash checking is forced for the duration of the switch. In this case, Execution halts when it encounters a modified or new statement in the worklist. If the return code low order 2 bit is on, the command that is executed under the switch is allowed to fail hash checking provided that the next command is not an SQL GRANT. This restriction means that the user can modify the worklist as needed, but cannot modify any GRANT statements. The low order 2 bit in the return code indicates whether or not to enforce hash failures when an SQL GRANT command is processed. Hash checking is enforced for the duration of the switch. In this case, Execution halts when it encounters a modified or new SQL GRANT statement in the worklist.</td>
</tr>
<tr>
<td>X’03’</td>
<td>termination/cleanup</td>
<td>immediately before termination</td>
<td>The call enables you to perform any necessary cleanup functions. For example, if you used another security product to perform security validations, you might want to sign off. Remember to perform a FREEMAIN to free any storage that you allocated. Return codes are ignored.</td>
</tr>
</tbody>
</table>
### Table 13  Execution security exit function codes (part 2 of 2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Call</th>
<th>Occurs</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'04'</td>
<td>SQL inspection</td>
<td>immediately before invoking dynamic SQL</td>
<td>The call enables you to inspect the SQL command that is about to be executed. You can use the command as it is, replace it with another command, or stop processing. If the AEXEFAUH bit of flag ASXEFLAG in the user exit DSECT is set, hash checking has failed on the -SQL command. The SQL call passes the address of the SQL statement in field ASXSQL. This address points to a field of the format $hh\ldots!x$, where $hh$ is a halfword length field, and $x\ldots!x$ is the SQL statement about to be executed. This field is passed to DB2 for execution. An SQL failure halts Execution.</td>
</tr>
<tr>
<td>X'05'</td>
<td>-SETS command inspection</td>
<td>immediately before executing a -SETS (SET CURRENT SQLID) worklist command</td>
<td>You can inspect, change, or deny the SQLID that is about to be switched. If the AEXEFAUH bit of flag ASXEFLAG in the user exit DSECT is set, hash checking has failed on the -AUTH command. This call does not affect hash checking because the SET CURRENT SQLID mechanism is inherently secured by the DB2 system itself. The primary purpose of this call is to allow the use of special SQLIDs that are not the creator or owner of DB2 objects that are being created in the worklist.</td>
</tr>
<tr>
<td>X'06'</td>
<td>-GLID command inspection</td>
<td>immediately before executing a -GLID (set global authorization ID) worklist command</td>
<td>You can inspect, change, or deny the global authorization ID (GLID) that is about to be switched. In addition, you can activate hash checking and other options on this exit call. If the AEXEFAUH bit of flag ASXEFLAG in the user exit DSECT is set, hash checking has failed on the -AUTH command.</td>
</tr>
</tbody>
</table>
Testing the Execution security exit

Following are some suggestions to help you develop and test your security exit:

1. Test the exit initially with simple logic and test cases to verify that you are correctly processing the function codes and return codes.

2. Add the DEBUG keyword to the AEXIN parameter input. Message BMC16990I is issued to AEXPRINT and includes the following information:
   - the function code with which the exit is called
   - the WORKID or SQLID that is about to be switched
   - the USERID that is associated with the job itself
   - the return code (Register 15) that is returned from the exit
   - the WORKID or SQLID that is returned from the exit

Contact BMC Customer Support if you have questions about the Execution security exit.

Restricting access to the worklist parallelism feature

With the Database Administration solution, you can use the worklist parallelism feature to execute portions of a worklist concurrently. CHANGE MANAGER uses the BMC Cross-System Image Manager (XIM) technology to provide sysplex performance improvements by enabling the distribution and management of discrete units of work (UOW) across one or more IBM OS/390 and z/OS images.

By default, user access to execute portions of a worklist concurrently and to dynamically start XIM is not restricted. You can control access to these functions for a user or a group of users by performing the following tasks:

1. Apply a zap.

2. If you are using the IBM Resource Access Control Facility (RACF®), specify a general resource profile.

NOTE

If you are using another security package that is compatible with the System Authorization Facility (SAF), contact BMC Customer Support.
Applying a zap

To enable the restriction of access to these functions, apply the following zap to the Execution function of CHANGE MANAGER:

<table>
<thead>
<tr>
<th>NAME</th>
<th>AEXPMAIN MAINRACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>VER</td>
<td>003E 47F0,C1D8</td>
</tr>
<tr>
<td>REP</td>
<td>003E 4700,0000</td>
</tr>
<tr>
<td>CHECKSUM</td>
<td>0916482E</td>
</tr>
</tbody>
</table>

Specifying a general resource profile

In RACF, general resource profiles are used to protect the resources that are defined in the class descriptor table, such as programs. To restrict a user’s or group’s access to each of the worklist parallelism functions, you must add general resource profiles with the following profile information:

- CLASS => FACILITY
- PROFILE => BMCACM.ssid.PARALLEL.object

The profile definition contains the following values:

- BMCACM specifies that the profile is for CHANGE MANAGER.
- ssid represents the name of the DB2 subsystem or a DB2 group attachment name (wildcard characters can be used to match one or more characters).
- PARALLEL represents the function that is secured.
- object represents the object or resource name that is secured.
  - For executing a worklist, the object is EXECUTE.
  - For starting XIM dynamically, the object is DYNSTART.

Each user or group that is given access to a resource profile must have an access level of CONTROL or higher.
Setting the MEMLIMIT system parameter

ALTER and CHANGE MANAGER require above-the-bar memory and might abend if sufficient memory is not available. The default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB. This value is set in member SMFPRMxx in SYS1.PARMLIB. Use any of the following methods if you need to override the default value:

- Specify the MEMLIMIT parameter in the JCL.
- Specify REGION=0M in the JCL.
- Use the SMF IEFUSI exit.

Modifying the ISPF skeletons and variables

The JCL Generation component uses standard ISPF file tailoring to generate all JCL. The $AJXDOC member in the HLQ.DBSLIB data set lists and briefly describes each nonDD and DD statement skeletons that JCL Generation uses. The member also lists the variables that JCL Generation uses to construct the default names for permanent data sets.

Each skeleton name in the $AJXDOC member has a corresponding member in the HLQ.DBSLIB data set. Some of the members that contain skeletons also provide comments, tips, and suggestions for using the skeletons.

You can specify up to five user-defined variables in the POF. The JCL Generation User Defined Variables panel allows you to specify variable names (up to eight characters). Each variable has a corresponding symbolic variable, as shown in Table 14. You can use the symbolic variables in your job cards or data set prefixes. For information about specifying the variables, see “Setting user variables” on page 213.

Table 14  User-defined variables

<table>
<thead>
<tr>
<th>POF variable</th>
<th>Symbolic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_VAR1_CHAR</td>
<td>&amp;UVR1</td>
</tr>
<tr>
<td>USER_VAR2_CHAR</td>
<td>&amp;UVR2</td>
</tr>
<tr>
<td>USER_VAR3_CHAR</td>
<td>&amp;UVR3</td>
</tr>
<tr>
<td>USER_VAR4_CHAR</td>
<td>&amp;UVR4</td>
</tr>
<tr>
<td>USER_VAR5_CHAR</td>
<td>&amp;UVR5</td>
</tr>
</tbody>
</table>

The ALTER and CHANGE MANAGER for DB2 Reference Manual lists and describes the symbolic and SLIB variables that JCL Generation uses and specifies the length of each variable.
To improve the performance of the JCL construction phase of JCL Generation, you can use an SLIB compiler. However, if you edit SLIBs after installation, you must recompile them. For information, see the ALTER and CHANGE MANAGER for DB2 User Guide.

Sizing the data sets

You can use the JCL Generation data set sizing function to tailor the data set sizes when the JCL is built. The function gathers information from one of the following sources (shown in general order of accuracy, from most accurate to least accurate):

1. the BMC DASD MANAGER PLUS product statistics database (statistics that the BMCSTATS utility gathers)
2. the DB2 system catalog (statistics that the IBM RUNSTATS utility gathers)
3. the results of VSAM object sampling
4. the default data set allocation parameters that are set from the JCL Generation Individual Data Set Options panel

Whether it uses the BMCSTATS historical database or object sampling, or even if it does not use sizing, JCL Generation obtains some information from the DB2 system catalog.

The formulas for estimating data set size are taken from the documentation for the IBM DB2 utilities and from the documentation for the BMC utilities. Table 15 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.

Table 15  Data set sizing values and sources (part 1 of 2)

<table>
<thead>
<tr>
<th>Value</th>
<th>BMCSTATS</th>
<th>IBM RUNSTATS</th>
<th>VSAM sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of active pages</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>number of modified pages</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>page size</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>maximum row length</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>average row length</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of rows</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>number of nonclustering indexes</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>longest key</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of foreign keys</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Maintaining the product tables

You are responsible for maintaining the tables that are used by ALTER and CHANGE MANAGER. In addition to installing the tables, maintenance tasks for the tables include

- sizing the tables to accommodate growth requirements
- purging the tables of old or unneeded data
- updating the rename table
- reorganizing the tables to improve performance
- backing up for recovery

For more information about the administrative tasks associated with managing the application databases, see the IBM documentation.

CHANGE MANAGER uses the tables that are listed in Table 17. ALTER uses all of the CD tables and some of the CM tables. For versions 8.3 and later, these tables store the version and release of the installed product. Earlier versions prefixed the table names with Vvr, where the vr reflected the product version and release.

The products use the prefixes listed in the default table name (Table 16).

### Table 15 Data set sizing values and sources (part 2 of 2)

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td>number of indexes</td>
<td>X</td>
</tr>
<tr>
<td>longest foreign key</td>
<td></td>
</tr>
</tbody>
</table>

**TIP**

To specify the sizing method, see “Setting the JCL options for static data sets” on page 184.

### Table 16 Table prefixes

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL</td>
<td>Baseline component of CHANGE MANAGER</td>
</tr>
<tr>
<td>CD</td>
<td>Specification and Analysis components of ALTER and CHANGE MANAGER</td>
</tr>
<tr>
<td>CM</td>
<td>objects for ALTER and CHANGE MANAGER</td>
</tr>
<tr>
<td>CP</td>
<td>CM/PILOT component of CHANGE MANAGER</td>
</tr>
<tr>
<td>Default table name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BL_ALIAS</td>
<td>alias definitions</td>
</tr>
<tr>
<td>BL_AUXTABLE</td>
<td>auxiliary object (such as LOB table space, auxiliary table, and auxiliary index) definitions</td>
</tr>
<tr>
<td>BL_CHECK</td>
<td>check constraint definitions</td>
</tr>
<tr>
<td>BL_CHECK2</td>
<td>path schema for check constraints</td>
</tr>
<tr>
<td>BL_COLAUTH</td>
<td>column authorizations</td>
</tr>
<tr>
<td>BL_COLUMN</td>
<td>table and view column definitions</td>
</tr>
<tr>
<td>BL_DATABASE</td>
<td>database definitions</td>
</tr>
<tr>
<td>BL_DBAUTH</td>
<td>security information for databases in this baseline</td>
</tr>
<tr>
<td>BL_INDEX</td>
<td>index definitions</td>
</tr>
<tr>
<td>BL_KEY</td>
<td>index key definitions</td>
</tr>
<tr>
<td>BL_KEYTARGETS</td>
<td>key-targets for extended index definitions</td>
</tr>
<tr>
<td>BL_REL</td>
<td>foreign key definitions</td>
</tr>
<tr>
<td>BL_RELKEY</td>
<td>foreign key column definitions</td>
</tr>
<tr>
<td>BL_RESAUTH</td>
<td>security information for table spaces and storage groups in this baseline</td>
</tr>
<tr>
<td>BL_SEQUENCES</td>
<td>identity column definitions</td>
</tr>
<tr>
<td>BL_STOGROUP</td>
<td>storage group definitions</td>
</tr>
<tr>
<td>BL_SYNONYM</td>
<td>synonym definitions</td>
</tr>
<tr>
<td>BL_TABAUTH</td>
<td>security information for tables and views in this baseline</td>
</tr>
<tr>
<td>BL_TABLE</td>
<td>table definitions</td>
</tr>
<tr>
<td>BL_TABLEPART</td>
<td>table and index space partition definitions</td>
</tr>
<tr>
<td>BL_TABSPACE</td>
<td>table space definitions</td>
</tr>
<tr>
<td>BL_TRIG_COL</td>
<td>trigger column definitions</td>
</tr>
<tr>
<td>BL_TRIG_TEXT</td>
<td>trigger text definitions</td>
</tr>
<tr>
<td>BL_TRIG_TEXT_AUX</td>
<td>trigger text definitions for auxiliary tables</td>
</tr>
<tr>
<td>BL_TRIGGER</td>
<td>trigger definitions</td>
</tr>
<tr>
<td>BL_UNQCNST</td>
<td>unique constraint definitions</td>
</tr>
<tr>
<td>BL_UNQCNST_COL</td>
<td>unique constraint column definitions</td>
</tr>
<tr>
<td>BL_VIEW</td>
<td>view definitions</td>
</tr>
<tr>
<td>BL_VIEW_TEXT</td>
<td>view text definitions</td>
</tr>
<tr>
<td>BL_VIEW_TEXT_AUX</td>
<td>view text definitions for auxiliary tables</td>
</tr>
<tr>
<td>BL_VOLS</td>
<td>additional volume information for storage groups and table partitions</td>
</tr>
<tr>
<td>CD_ALIAS</td>
<td>alias definitions</td>
</tr>
<tr>
<td>CD_AUXTABLE</td>
<td>auxiliary object (such as LOB table space, auxiliary table, and auxiliary index) definitions</td>
</tr>
<tr>
<td>CD_CHECK</td>
<td>check constraint definitions</td>
</tr>
<tr>
<td>CD_CHECK2</td>
<td>path schema for check constraints</td>
</tr>
<tr>
<td>CD_COLUMN</td>
<td>table and view column definitions</td>
</tr>
</tbody>
</table>
### Table 17  ALTER and CHANGE MANAGER tables (part 2 of 3)

<table>
<thead>
<tr>
<th>Default table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD_DATABASE</td>
<td>database definitions</td>
</tr>
<tr>
<td>CD_INDEX</td>
<td>index definitions</td>
</tr>
<tr>
<td>CD_KEY</td>
<td>index key definitions</td>
</tr>
<tr>
<td>CD_KEYTARGETS</td>
<td>key-targets for extended index definitions</td>
</tr>
<tr>
<td>CD_REL</td>
<td>foreign key definitions</td>
</tr>
<tr>
<td>CD_RELKEY</td>
<td>foreign key column definitions</td>
</tr>
<tr>
<td>CD_ROUTINES</td>
<td>stored procedure definitions</td>
</tr>
<tr>
<td>CD_SEQUENCES</td>
<td>identity column definitions</td>
</tr>
<tr>
<td>CD_STOGROUP</td>
<td>storage group definitions</td>
</tr>
<tr>
<td>CD_SYNONYM</td>
<td>synonym definitions</td>
</tr>
<tr>
<td>CD_TABLE</td>
<td>table definitions</td>
</tr>
<tr>
<td>CD_TABLEPART</td>
<td>table and index space partition definitions</td>
</tr>
<tr>
<td>CD_TABLESPACE</td>
<td>table space definitions</td>
</tr>
<tr>
<td>CD_TRIG_COL</td>
<td>trigger column definitions</td>
</tr>
<tr>
<td>CD_TRIG_TEXT</td>
<td>trigger text definitions</td>
</tr>
<tr>
<td>CD_TRIG_TEXT_AUX</td>
<td>trigger text definitions for auxiliary tables</td>
</tr>
<tr>
<td>CD_TRIGGER</td>
<td>trigger definitions</td>
</tr>
<tr>
<td>CD_UNQCNST</td>
<td>unique constraint definitions</td>
</tr>
<tr>
<td>CD_UNQCNST_COL</td>
<td>unique constraint column definitions</td>
</tr>
<tr>
<td>CD_VIEW</td>
<td>view definitions</td>
</tr>
<tr>
<td>CD_VIEW_TEXT</td>
<td>view text definitions</td>
</tr>
<tr>
<td>CD_VIEW_TEXT_AUX</td>
<td>view text definitions for auxiliary tables</td>
</tr>
<tr>
<td>CD_VOLS</td>
<td>additional volume information for storage groups and table parts</td>
</tr>
<tr>
<td>CM_BASELINE</td>
<td>baseline header information</td>
</tr>
<tr>
<td>CM_BLPROFILE</td>
<td>baseline profile information</td>
</tr>
<tr>
<td>CM_LOCATION</td>
<td>location information for migrate profiles</td>
</tr>
<tr>
<td>CM_MIGPROFILE</td>
<td>migrate profile information</td>
</tr>
<tr>
<td>CM_RENAME</td>
<td>old and new names of renamed objects</td>
</tr>
<tr>
<td>CM_RULES</td>
<td>rules for migrate profiles</td>
</tr>
<tr>
<td>CM_SCOPE</td>
<td>scope for migrate and baseline profiles</td>
</tr>
<tr>
<td>CM_SECURITY</td>
<td>security table</td>
</tr>
<tr>
<td>CM_SYNC</td>
<td>execution sync table</td>
</tr>
<tr>
<td>CM_UNLOADDSN</td>
<td>information about unloaded data sets used in a full-recovery baseline</td>
</tr>
<tr>
<td>CM_WORKID</td>
<td>information about each work ID</td>
</tr>
<tr>
<td>CP_APPLICATION</td>
<td>information about each CM/PILOT application</td>
</tr>
<tr>
<td>CP_DML_STMT</td>
<td>information about each DML statement in each CM/PILOT task ID</td>
</tr>
<tr>
<td>CP_SCRIPT</td>
<td>information about user-defined CM/PILOT scripts</td>
</tr>
<tr>
<td>CP_STEP</td>
<td>information about steps in user-defined CM/PILOT scripts</td>
</tr>
</tbody>
</table>
Maintaining the size of tables

The baseline tables (those named BL\_name) of CHANGE MANAGER require more ongoing maintenance than other tables. Because the baseline tables hold critical information used for database recovery, information in them should be retained as long as the baselines are needed. If your installation regularly establishes baselines, these tables grow over time. You must ensure that these tables are reorganized and that the size of the tables is expanded on a regular basis. To help control the baseline table, you can delete any unwanted baselines. For more information about deleting baselines, see the ALTER and CHANGE MANAGER for DB2 User Guide.

The sync tables of ALTER and CHANGE MANAGER (CM\_SYNC) and the CM/PILOT component (CP\_SYNC) also require more ongoing maintenance. The records in the sync table for a particular work ID or task ID are used as progress indicators during execution of a worklist. These tables grow over time. You will need to ensure that these tables are reorganized and expanded on a regular basis. You will also need to delete any work IDs and task IDs that are no longer needed.

To help control the size of the sync tables, you can delete any unwanted work IDs or task IDs. When you delete a work ID, all of the change and migration requests for that work ID are deleted from the product’s CD tables. If the status of the work ID is Exec Strt (for Execution started) or Exec Comp (for Execution complete), all of the rows that are written to the product’s work ID tables (such as the sync table) are also deleted. Deleting the work ID from the product tables does not affect committed changes that were made to the DB2 catalog during execution of the work ID. (For information about viewing the status of the worklist, see the ALTER and CHANGE MANAGER for DB2 User Guide.)

NOTE
Deleting a work ID with Exec Strt status is not recommended because the work ID cannot be restarted for completion.

### Table 17  ALTER and CHANGE MANAGER tables (part 3 of 3)

<table>
<thead>
<tr>
<th>Default table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP_SYNC</td>
<td>sync table for CM/PILOT worklist execution</td>
</tr>
<tr>
<td>CP_TASKID</td>
<td>information about each CM/PILOT task ID</td>
</tr>
</tbody>
</table>
Securing tables

Only those users who need to perform diagnostic functions on ALTER or CHANGE MANAGER should be given access to the ALTER or CHANGE MANAGER database objects. Access to the database through ALTER and CHANGE MANAGER functions is achieved through EXECUTE authority on plans, except on the rename table.

The Compare component uses the CHANGE MANAGER rename table (CM_RENAME) when a baseline is used as input to resolve the names of renamed objects. If users change the names of DB2 objects outside of CHANGE MANAGER, they should update the CHANGE MANAGER rename table so that the Compare component can correctly resolve renamed objects. To update the table, the users must submit SQL INSERT statements through a DB2 access tool, and must therefore have authority to update the rename table. Table 18 shows the SQL INSERT statements for manually updating the CHANGE MANAGER rename table. Access to the rename table is available by granting DB2 authorization to INSERT, UPDATE, DELETE and SELECT that table. For more information about renamed object resolution, see the ALTER and CHANGE MANAGER for DB2 User Guide.

WARNING
Failure to properly update the rename table could result in data loss, because Compare generates DROP and CREATE statements rather than an ALTER statement for renamed objects.

WARNING
Full-recovery baselines are associated with alter-type work IDs. If you delete a work ID that has an associated full-recovery baseline, the work ID for that baseline will become null.

TIP
To delete work IDs or task IDs, see the ALTER and CHANGE MANAGER for DB2 User Guide.
<table>
<thead>
<tr>
<th><strong>DB2 object</strong></th>
<th><strong>INSERT statement</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>storage group</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NEWNAME_PART1, TIMESTMP) VALUES ('SG', 'oldName', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>database</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NEWNAME_PART1, TIMESTMP) VALUES ('DB', 'oldName', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>table space</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('TS', 'oldDatabase', 'oldTableSpace', 'newDatabase', 'newTableSpace', 'timeStamp')</td>
</tr>
<tr>
<td>table</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('TB', 'oldCreator', 'oldName', 'newCreator', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>table column</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, CONAME, NEWNAME_PART1, NEWNAME_PART2, NEWCONAME, TIMESTMP) VALUES ('CO', 'oldCreator', 'oldTable', 'oldColumn', 'newCreator', 'newTable', 'newColumn', 'timeStamp')</td>
</tr>
<tr>
<td>check constraint</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NAME_PART3, NEWNAME_PART1, NEWNAME_PART2, NEWNAME_PART3, TIMESTMP) VALUES ('CK', 'oldCreator', 'oldTable', 'oldName', 'newCreator', 'newTable', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>index</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('IX', 'oldCreator', 'oldName', 'newCreator', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>unique constraint</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NAME_PART3, NEWNAME_PART1, NEWNAME_PART2, NEWNAME_PART3, TIMESTMP) VALUES ('UC', 'oldCreator', 'oldTable', 'oldName', 'newCreator', 'newTable', 'newName', 'newTimeStamp')</td>
</tr>
</tbody>
</table>
Table 18  SQL for updating rename table (part 2 of 2)

<table>
<thead>
<tr>
<th>DB2 object</th>
<th>INSERT statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>trigger</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('TR','oldSchema','oldName', 'newSchema', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>foreign key</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NAME_PART3, NEWNAME_PART1, NEWNAME_PART2, NEWNAME_PART3, TIMESTMP) VALUES ('FK','oldCreator','oldTable', 'oldName', 'newCreator', 'newTable', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>view</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('VW','oldCreator','oldName', 'newCreator', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>view column</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, CONAME, NEWNAME_PART1, NEWNAME_PART2, NEWCONAME, TIMESTMP) VALUES ('VC','oldCreator','oldView', 'oldColumn', 'newCreator', 'newView', 'newColumn', 'timeStamp')</td>
</tr>
<tr>
<td>synonym</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('SY','oldCreator','oldName', 'newCreator', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>alias</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('AL','oldCreator','oldName', 'newCreator', 'newName', 'timeStamp')</td>
</tr>
<tr>
<td>sequence</td>
<td>INSERT INTO CM_RENAME (OBJECT, NAME_PART1, NAME_PART2, NEWNAME_PART1, NEWNAME_PART2, TIMESTMP) VALUES ('SQ','oldSchema','oldName', 'newSchema', 'newName', 'timeStamp')</td>
</tr>
</tbody>
</table>
Table 19 lists the tasks that you can perform to set your user and product options in ALTER or CHANGE MANAGER.

**Table 19  Set up tasks (part 1 of 2)**

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting ALTER and CHANGE MANAGER options</td>
<td></td>
</tr>
<tr>
<td>“Setting the attributes for the panel display”</td>
<td>page 164</td>
</tr>
<tr>
<td>“Setting the options for data set allocation”</td>
<td>page 166</td>
</tr>
<tr>
<td>“Setting the processing options for Analysis”</td>
<td>page 167</td>
</tr>
<tr>
<td>“Setting the processing options for Import”</td>
<td>page 169</td>
</tr>
<tr>
<td>“Setting the processing options for Execution”</td>
<td>page 171</td>
</tr>
<tr>
<td>Setting CHANGE MANAGER and CM/PILOT options</td>
<td></td>
</tr>
<tr>
<td>“Setting the processing options for Compare”</td>
<td>page 172</td>
</tr>
<tr>
<td>“Setting the processing options for Baseline”</td>
<td>page 174</td>
</tr>
<tr>
<td>“Setting the options for the baseline report”</td>
<td>page 175</td>
</tr>
<tr>
<td>“Setting the user options for CM/PILOT”</td>
<td>page 176</td>
</tr>
<tr>
<td>Setting JCL Generation options</td>
<td></td>
</tr>
<tr>
<td>“Setting the JCL options for job cards”</td>
<td>page 180</td>
</tr>
<tr>
<td>“Setting the JCL options for STEPLIBs”</td>
<td>page 182</td>
</tr>
<tr>
<td>“Setting the JCL options for static data sets”</td>
<td>page 184</td>
</tr>
<tr>
<td>“Setting the JCL options for tapes”</td>
<td>page 188</td>
</tr>
<tr>
<td>“Setting the JCL options for temporary work data sets”</td>
<td>page 191</td>
</tr>
<tr>
<td>“Setting the JCL options for permanent data sets”</td>
<td>page 193</td>
</tr>
<tr>
<td>“Setting the JCL generation data group options”</td>
<td>page 202</td>
</tr>
<tr>
<td>“Setting the JCL debugging, display, and Execution options”</td>
<td>page 203</td>
</tr>
<tr>
<td>“Setting the JCL utility installation options module name options”</td>
<td>page 207</td>
</tr>
<tr>
<td>“Setting the online reorg options”</td>
<td>page 208</td>
</tr>
<tr>
<td>“Setting the Execution worklist parallelism options”</td>
<td>page 209</td>
</tr>
<tr>
<td>“Setting the worklist options”</td>
<td>page 211</td>
</tr>
<tr>
<td>“Setting user variables”</td>
<td>page 213</td>
</tr>
<tr>
<td>“Creating a user POF”</td>
<td>page 214</td>
</tr>
<tr>
<td>“Updating a user POF”</td>
<td>page 216</td>
</tr>
<tr>
<td>“Using multiple POFs”</td>
<td>page 217</td>
</tr>
<tr>
<td>“Refreshing the initial POF”</td>
<td>page 218</td>
</tr>
<tr>
<td>“Generating POF reports”</td>
<td>page 219</td>
</tr>
<tr>
<td>“Reusing a POF in a subsequent installation”</td>
<td>page 219</td>
</tr>
</tbody>
</table>
Setting the attributes for the panel display

To define or modify the values in your ISPF profile, you can use the Options panels of ALTER or CHANGE MANAGER. Use the Panel Attribute Options panel to specify your preferences for long names, entry field and error delimiters, and screen colors.

1. From the ALTER or CHANGE MANAGER Main Menu, select **Options** and press Enter.

2. From the ALTER or CHANGE MANAGER Main Options Menu panel, select **Panel Attribute Options** and press Enter.

The ALTER or CHANGE MANAGER Panel Attribute Options panel is displayed (Figure 57).

**Figure 57  Panel Attribute Options panel**

```
ALUFOPT1 ---------- CHANGE MANAGER Panel Attribute Options --------------
Command =>

Type information. Then press Enter to continue or PF12 for previous panel.

Zoom/Truncation Attributes.
  Allow popup displays . . . Y (Y/N)
  Omission Characters . . . <>
  Autotab Character . . . . +
  Location . . . . . . . . M (Beginning, Middle, End)

Select your preference for an entry field delimiter.
  1. Underscore(USCORE)  2. Reverse video  3. None

Specify color/delimiter to be used when an error is detected.
  Color of field . . . . .
  Delimiter . . . . . . . 2 1. Underscore  2. Reverse video  3. None

ISPF Color Attributes.
  Type any color preference for the color attribute you wish to substitute

ISPF Color Attributes       Color Attribute You Wish to Substitute
  Blue . . . . . . . . . .
  Green . . . . . . . . .

Commands:  RESET HELP END
```
3 Specify the values that should be used when displaying long names:

--- NOTE ---
The product stores the values for each of the zoom and truncation attributes in your ISPF profile. However, you can also write the values to a user POF.

A In the **Allow popup displays** field, specify whether to display a dialog or a panel when the ZOOM (F4) key is pressed on an object name:

- If you specify Y, the product displays the object name in a dialog. If the name is too long to be displayed in a dialog, the product displays the name in a panel.

- If you specify N, the product displays the object name in a panel.

B In the **Omission Characters** field, type the characters that replace the beginning and end of a truncated string in an object name that is too long to be displayed.

C In the **Autotab Character** field, specify whether to display an autotab character in front of an object name that is too long to be displayed:

- If you specify N, an autotab character is not displayed and you cannot tab to the object name. To zoom on the object name, position the cursor on the object name and press the ZOOM (F4) key.

- If you specify any character other than N, the character is displayed and you can tab to the object name. To zoom on the object name, press the ZOOM (PF4) key.

--- TIP ---
The autotab character indicates which fields must be zoomed to view the entire object name.

D In the **Location** field, type B, M, or E to specify the location of characters to be omitted in object names that are too long to be displayed.

<table>
<thead>
<tr>
<th>To replace characters</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>at the left end (beginning) of the name</td>
<td>B</td>
</tr>
<tr>
<td>in the middle of the name</td>
<td>M</td>
</tr>
<tr>
<td>at the right end (end) of the name</td>
<td>E</td>
</tr>
</tbody>
</table>

4 Type a value for your entry field delimiter.
Setting the options for data set allocation

To define or modify the values in your ISPF profile, you can use the Options panels of ALTER or CHANGE MANAGER. Use the Dataset Allocation Options and the Image Copy and Unload Options panels to specify the parameters for allocating your data sets. The worklist and CDL data sets are the only data sets that ALTER and CHANGE MANAGER allocate. All other data sets, such as JCL, Compare, Baseline, Report, and DDL, must be preallocated.

1. From the ALTER or CHANGE MANAGER Main Menu, select Options and press Enter.

The ALTER or CHANGE MANAGER Main Options Menu panel is displayed.

2. Select Dataset Allocation Options and press Enter.

The ALTER or CHANGE MANAGER Dataset Allocation Options panel is displayed, as shown in Figure 58.

**Figure 58  Dataset Allocation Options panel**

```
ALUFOPT2 ---------- CHANGE MANAGER Dataset Allocation Options ---------------
Command ===> 
Type information. Then press Enter to continue or PF12 for previous panel.
DB2 Subsystem ID .. DEGA

Specify Dataset Allocation Parameters.
Dataset Unitname Volume Priqty Secqty Alloc Unit
Worklist/BL Report SYSDA 15 5 Tracks
CDL SYSDA 15 5 Tracks

Commands: HELP END
```
3 Specify the DB2 subsystem ID (SSID).

4 Specify the data set allocation parameters for a worklist and a baseline report by typing new values for the Unitname, Volume, Priqty, and Secqty fields.

**NOTE**
The product stores the values for the data set allocation parameters in your ISPF profile. However, you can also write the values to a user POF.

5 *(CHANGE MANAGER only)* Specify the data set allocation parameters for the CDL data set by typing new values for the Unitname, Volume, Priqty, and Secqty fields.

**NOTE**
The product stores the values for the data set allocation parameters in your ISPF profile. However, you can also write the values to a user POF.

6 Press END to save your changes and to return to the ALTER or CHANGE MANAGER Main Menu.

### Setting the processing options for Analysis

To define or modify the values in your ISPF profile, you can use the Options panels of ALTER or CHANGE MANAGER. Use the Analysis Options panel to specify the data set names, processing options, and job parameters for the Analysis component.

1 From the ALTER or CHANGE MANAGER Main Menu, select Options and press Enter.

The ALTER or CHANGE MANAGER Main Options Menu panel is displayed.

2 Select Analysis Options and press Enter.

The ALTER or CHANGE MANAGER Analysis Options panel is displayed, as shown in Figure 59.
Figure 59  Analysis Options panel

3 Specify the Analysis data set names for the JCL, worklist, and diagnostics.

NOTE
The data sets for the JCL and worklist can be either sequential files or PDS members. The data set for the diagnostics must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

The values for these options are stored in your ISPF profile. However, the values can also be written to a user POF.

4 Specify the default processing options for Analysis by typing S to select an option.

5 Select whether to run Analysis in foreground or in batch.

6 Specify the default job parameters for Analysis.

NOTE
The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.
Setting the processing options for Import

7 (optional) Type a name for a default global AUTHID. For more information about global authorization IDs, see the *ALTER and CHANGE MANAGER for DB2User Guide*.

**NOTE**
The product stores the value for this option in your ISPF profile. However, you can also write the value to a user POF.

8 Press END to save your changes and to return to the ALTER or CHANGE MANAGER Main Menu.

### Setting the processing options for Import

To define or modify the values in your ISPF profile, you can use the Options panels of ALTER or CHANGE MANAGER. Use the Import Options panel to specify the data set names and the processing options for the Import component of ALTER or CHANGE MANAGER.

1 From the ALTER or CHANGE MANAGER Main Menu, select **Options** and press **Enter**.

   The ALTER or CHANGE MANAGER Main Options Menu panel is displayed.

2 Select **Import Options** and press **Enter**.

   The ALTER or CHANGE MANAGER Import Options panel is displayed, as shown in **Figure 60**.
3 Specify the Import data set names for the JCL and diagnostics.

**NOTE**
The JCL data set can be either a sequential file or a PDS member. The data set for the Diagnostics file must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.

4 Specify the default processing options for Import by typing **S** to select an option.

**NOTE**
In ALTER, you can edit the DDL. In CHANGE MANAGER, you can edit the CDL or DDL.

5 Specify whether to run Import in foreground or in batch.

6 Press END to save your changes and to return to the ALTER or CHANGE MANAGER Main Menu.
Setting the processing options for Execution

To define or modify the values in your ISPF profile, you can use the Options panels of ALTER or CHANGE MANAGER. Use the Execution Options panel to specify the data set names and the method of generating JCL for the Execution component.

1. From the ALTER or CHANGE MANAGER Main Menu, select Options and press Enter.

   The ALTER or CHANGE MANAGER Main Options Menu panel is displayed.

2. Select Execution Options and press Enter.

   The Execution Options panel is displayed, as shown in Figure 61.

   **Figure 61  Execution Options panel**

   ```
   ALUFOPTA  ----------- CHANGE MANAGER Execution Options  -----------
   Command ===>
   Type information. Then press Enter to continue or PF12 for previous panel.

   Specify EXECUTION dataset names and optional DBRM Libraries.
   Execution JCL . . . 'ACM.DB2V10.V10.EXECJCL(&WORKID8)'
   Diagnostics . . . SYSOUT
   Batch JCL Job . . . 'ACM.DB2V10.V10.BATCH.EXECJCL(&WORKID8)'
   DBRM Concat'd 1st .
   DBRM Concat'd 2nd .
   DBRM Concat'd 3rd .

   Select default EXECUTION processing options.
   _ Override Options
   _ Edit Worklist
   S Build JCL
   S Edit JCL
   _ Submit
   _ Save Last Used Options

   Select JCL Generation Option  1 1. Foreground  2. Batch
   Commands:  HELP END
   ```

3. Specify the Execution default data set names for the JCL and diagnostics.

   The Execution JCL contains the JCL to run Execution. The batch JCL job contains the JCL to run Batch Execution JCL Generation, which builds the JCL to run Execution.
Setting the processing options for Compare

To define or modify the values in your ISPF profile, you can use the Options panels of CHANGE MANAGER. Use the Compare Options panel to specify the data set names and the processing options for the Compare component of CHANGE MANAGER.

1. From the CHANGE MANAGER Main Menu, select Options and press Enter.
   
   The CHANGE MANAGER Main Options Menu panel is displayed.

2. Select Compare Options and press Enter.
   
   The CHANGE MANAGER Compare Options panel is displayed, as shown in Figure 62.

   **NOTE**
   
The data set for JCL can be either a sequential file or a PDS member.

   The data set for the Diagnostics file must be a sequential file, unless parallelism is enabled for the Database Administration solution. If you are running a worklist in parallel, the diagnostics file must be allocated to SYSOUT. AEXPRINT is the ddname of the diagnostic output.

   The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.

4. (optional) Specify up to three Database Request Module (DBRM) libraries that can be concatenated in the Execution JCL for binds in a worklist.

   **NOTE**
   
The product stores the values for these options in your ISPF profile. However, you can write the values to a user POF.

5. Specify the default processing options for Execution by typing S to select an option.

6. Select whether to generate JCL in foreground or in batch.

7. Press END to save your changes and to return to the ALTER or CHANGE MANAGER Main Menu.
3 Specify the Compare data set names for the CDL, JCL, and diagnostics.

**NOTE**

The data sets for JCL and the CDL file can be either sequential files or PDS members. The data set for the Diagnostics file must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

The product stores the values for the CDL, JCL, and diagnostics data sets in your ISPF profile. However, you can also write the values to a user POF.

4 Specify whether to generate comparison report information as comments in the CDL file.

5 Specify the default processing options for Compare by typing S to select an option.

6 Specify whether to run Compare in foreground or in batch.

7 Press END to save your changes and to return to the CHANGE MANAGER Main Menu.
Setting the processing options for Baseline

To define or modify the values in your ISPF profile, you can use the Options panels of CHANGE MANAGER. Use the Baseline Options panel to specify the data set names and the processing options for the Baseline component of CHANGE MANAGER.

1. From the CHANGE MANAGER Main Menu, select **Options** and press **Enter**.

   The CHANGE MANAGER Main Options Menu panel is displayed.

2. Select **Baseline Options** and press **Enter**.

   The CHANGE MANAGER Baseline Options panel is displayed, as shown in Figure 63.

   **Figure 63  Baseline Options panel**

   ```
   ALUFOPT6  --------------- CHANGE MANAGER Baseline Options ------------------------
   Command ===>

   Type information. Then press Enter to continue or PF12 for previous panel.

   Specify BASELINE dataset names.
   JCL . . . . . . 'RDACRJ.V10.BLJCL(&WORKID8)'
   Diagnostics . . SYSOUT

   Select default BASELINE processing options.     Run Type . . 2 1. Foreground
   S Build JCL/Parms
   S Edit JCL/Parms
   _ Submit/Run
   _ Save Last Used Options

   Commands:  HELP END
   ```

3. Specify the Baseline data set names for the JCL and diagnostics.

   **NOTE**

   The JCL data set can be either a sequential file or a PDS member. The data set for the Diagnostics file must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

   The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.
4 Specify the default processing options for Baseline by typing S to select an option.

5 Specify whether to run Baseline in foreground or in batch.

6 Press END to save your changes and to return to the CHANGE MANAGER Main Menu.

---

**Setting the options for the baseline report**

To define or modify the values in your ISPF profile, you can use the Options panels of CHANGE MANAGER. Use the Baseline Report Options panel to specify the data set names and the processing options for the baseline report for CHANGE MANAGER.

1 From the CHANGE MANAGER Main Menu, select Options and press Enter.

   The CHANGE MANAGER Main Options Menu panel is displayed.

2 Select Baseline Report Options and press Enter.

   The CHANGE MANAGER Baseline Report Options panel is displayed, as shown in Figure 64.

**Figure 64  Baseline Report Options panel**

```
ALUFOPT7 ----------- CHANGE MANAGER Baseline Report Options -------------------
Command ===>

Type information. Then press Enter to continue or PF12 for previous panel.

Specify BASELINE REPORT dataset names.
JCL . . . . . . 'RDACRJ.V10.BLRJCL(&WORKID8)'
Report . . . . 'RDACRJ.V10.BLREPS(&WORKID8)'
Diagnostics . . SYSOUT

Select default REPORT processing options. Run Type . . 1 1. Foreground
  S Build JCL/Parms
  S Edit JCL/Parms
_ Submit/Run
  S Edit Report
_ Save Last Used Options

Commands:  HELP END
```
3 Specify the baseline report data set names for the JCL, report, and diagnostics.

**NOTE**
The JCL data set can be either a sequential file or a PDS member. The data set for the Diagnostics file must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.

4 Specify the default processing options for the baseline report by typing $ to select an option.

5 Specify whether to run the baseline report in foreground or in batch.

6 Press END to save your changes and to return to the CHANGE MANAGER Main Menu.

**Setting the user options for CM/PILOT**

The first time that you run CHANGE MANAGER, the Front End copies the values from the installation options module into your ISPF profile. To define or modify the values in your ISPF profile, you can use the Options panels of the CM/PILOT component of CHANGE MANAGER. Use the panels to specify the task ID data set names, Worklist processing options, application profile names, and CHANGE MANAGER data set names.

1 From the CHANGE MANAGER Main Menu, select CM/PILOT and press Enter.

The CM/PILOT Main Menu is displayed.

2 From the CM/PILOT Main Menu, select Options and press Enter. The CM/PILOT TASKID Options panel is displayed, as shown in Figure 65.
3 Specify the task ID data set names for the worklist, JCL, and diagnostics.

These data set names appear as defaults on the TASKID Interface panel.

NOTE
The JCL data set can be either a sequential file or a PDS member. The data set for the Diagnostics file must be a sequential file. Diagnostics might also be sent to SYSOUT or to TERM (in foreground).

The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.

4 Specify the default processing options for the CM/PILOT worklist by typing S to select an option.

5 Press Enter to display the CM/PILOT Application Options panel, which is shown in Figure 66.
6 Specify the profile names and a name template for the work ID. These names appear as defaults on the Create Application panel.

- Type the name of the **Inbound Migrate Profile** that is used by the Import component of CHANGE MANAGER for applying change rules to attributes of data structures that are imported to the local DB2 subsystem.

- Type the name of the **Outbound Migrate Profile** that is used by the Analysis and Compare components of CHANGE MANAGER for generating worklists and CDL files. Outbound migrate profiles can contain change rules, scope rules, and locations.

- Type the name of the **DDL Baseline Profile** that is used by the Baseline component of CHANGE MANAGER for creating baselines of DDL files and worklists.

- Type the name of the **Catalog Baseline Profile** that is used by the Analysis, Baseline, and Compare components of CHANGE MANAGER for specifying the data structures from the DB2 catalog to include in a baseline.

**NOTE**

You can specify wildcard patterns for the profiles on the CM/PILOT Application Options panel. Then, when you create an application, the wildcard patterns appear on the Create Application panel. When you press **Enter** on the Create Application panel, a list of profiles is displayed.
Type the name of the **WORKID Name Template** that enables you to standardize naming conventions for new work IDs that CM/PILOT creates.

**7** Press **Enter** to display the CM/PILOT Output Dataset Options panel, which is shown in **Figure 67**.

**Figure 67**  **CM/PILOT Output Dataset Options panel**

```
AUCFOPT3 ----------------- CM/PILOT Output Dataset Options -----------------
Command ===> Type information. Then press Enter to continue or PF12 for previous panel.

Specify CHANGE MANAGER dataset names.
  JCL . . . . . . 'RDACRJ.V10.EXECJCL(&WORKID8)'
  Worklist . . . 'RDACRJ.V10.WLBASED1(&WORKID8)'
  CDL . . . . . . 'RDACRJ.V10.CDLBASED1(&WORKID8)'

Commands: HELP END CANCEL
```

8 Specify the CHANGE MANAGER data set names. These data set names appear as CHANGE MANAGER defaults on the CHANGE MANAGER Datasets panel.

**NOTE**
You cannot modify the CM/PILOT user options for the CHANGE MANAGER data set names by changing the user options in CHANGE MANAGER. You can modify these user options only from the CM/PILOT Output Dataset Options panel or by refreshing the user options.

The product stores the values for these options in your ISPF profile. However, you can also write the values to a user POF.

- Type the name of the JCL data set that contains the JCL to process the Analysis worklist.
- Type the name of the worklist data set that contains the Analysis worklist.
- Type the name of the CDL data set that contains the CDL which results from a comparison.
To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Jobcard Options panel to specify information about the job cards used in the JCL.

**To set the JCL options for job cards**

1. Use the following menu selections to display the JCL Generation Jobcard Options Update panel (Figure 68):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

**Figure 68**  JCL Generation Jobcard Options Update panel

AJXOJOBP ---------- JCL GENERATION JOBCARD OPTIONS UPDATE ----------------------
COMMAND ===> Type data and press Enter.

Is a TSO submit exit used to generate jobcards? . . N  (Y/N)

Enter Jobcards below:
//&USERID.&JOBCHAR JOB (&ZACCTNUM),.'&PGMR',
  // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
  // NOTIFY=&USERID
/*ROUTE XEQ BMCPLX1
/*JOBPARM SYSAFF=&ZSYSID
Jcllib . . . . . . .
Sysexec. . . . . . .
Region size . . . . 0M (See JCL Reference for valid options)
Memlimit . . . . . . (See JCL Reference for valid options)
Time parameter . . (See JCL Reference for valid options)
System MLIB . . . ISP.SISPMENU
Runtime HLQ. . . . AEX.QA101
LLQ. . . . . . . . . BMC
ULLQ . . . . . . . . (Leave blank if not using runtime enablement)
2 In the Is a TSO submit exit used to generate jobcards? field, type Y or N.

3 At Enter Jobcards below, type the job statement information that you want to add to the JCL.

4 In the Jcllib field, type the name of a PDS that contains customized JCL to be included in a job.

5 In the Sysexec field, type the name of the partitioned data set in which a REXX EXEC is a member.

**TIP**

To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.

6 In the Region size field, type the amount of memory to allocate for each step so that your job can run.

7 In the Memlimit field, type the limit on the above-the-bar memory for an address space.

8 In the Time parameter field, type the value for the TIME limit for each step in a batch job stream.

9 In the System MLIB field, type the name of the system ISPF message library.

10 In the Runtime HLQ field, type the high-level qualifier (HLQ) for ISPF data sets for the installation environment.

During installation, if you chose to use the runtime enablement (RTE) feature, the Installation System set this value to an HLQ for user runtime libraries. If you chose not to use RTE, the Installation System set the value to an HLQ for Execution.

This value supports the following symbolic variables:

- &DB2V2 and &DB2V3, which resolve to the version of DB2
- &SSID, which resolves to the DB2 subsystem ID

When you include the &SSID symbolic variable, the product can use a single POF with multiple subsystems.
11 In the LLQ field, type the low-level qualifier for ISPF data sets for the installation environment.

During installation, if you chose to use the runtime enablement feature, the Installation System set this value to BMC. If you chose not to use the feature, the Installation System set the value to DB.

12 In the ULLQ field, type the low-level qualifier for user-defined data sets for the installation environment.

During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not change the value. If you chose not to use the feature, the Installation System set the value to UDB.

13 Press END to save your changes and return to the ALTER or CHANGE MANAGER Main Menu.

Setting the JCL options for STEPLIBs

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the STEPLIB Options panel to specify the load libraries that appear in jobs that ALTER or CHANGE MANAGER creates.

**To set the JCL options for STEPLIBs**

1 Use the following menu selections to display the JCL Generation STEPLIB Options Update panel (Figure 69):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Steplib Options</td>
</tr>
</tbody>
</table>
Figure 69  JCL Generation STEPLIB Options Update panel

<table>
<thead>
<tr>
<th>AJXOSTP</th>
<th>JCL GENERATION STEPLIB OPTIONS UPDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>====&gt;</td>
</tr>
<tr>
<td>Type data and press Enter.</td>
<td></td>
</tr>
<tr>
<td>DSNEXIT . . . . . . . . SY3.DBDC.DSNEXIT</td>
<td></td>
</tr>
<tr>
<td>DB2 DSNLOAD . . . . . . SYS2.DB2V10M.DSNLOAD</td>
<td></td>
</tr>
<tr>
<td>Override lib . . . . . .</td>
<td></td>
</tr>
<tr>
<td>CATALOG MANAGER . . . .</td>
<td></td>
</tr>
<tr>
<td>ALTER/CHANGE MANAGER . .</td>
<td></td>
</tr>
<tr>
<td>DASD MANAGER PLUS . . . .</td>
<td></td>
</tr>
<tr>
<td>EXECUTION . . . . . . . 'BMCADMN.V101.D10.BMCLINK'</td>
<td></td>
</tr>
<tr>
<td>COPY PLUS . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>REORG PLUS . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>LOADPLUS . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>UNLOAD PLUS . . . . . .</td>
<td></td>
</tr>
<tr>
<td>RECOVER PLUS . . . . . .</td>
<td></td>
</tr>
<tr>
<td>CHECK PLUS . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>SQL EXPLORER . . . . . .</td>
<td></td>
</tr>
<tr>
<td>Additional lib . . . . .</td>
<td></td>
</tr>
<tr>
<td>IOA LOAD 1 . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>IOA LOAD 2 . . . . . . .</td>
<td></td>
</tr>
</tbody>
</table>

2 Specify the data set names for the libraries:

**TIP**
To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.

**A** In the DSNEXIT field, type the data set name of the DB2 EXIT (DSNEXIT) library.

**TIP**
To indicate the version of DB2, append the &DB2V2 or &DB2V3 symbolic variable to the name.

**B** In the DB2 DSNLOAD field, type the data set name of the library in which the DB2 DSN Command Processor load modules are stored.

**C** In the Override lib field, type the data set name for the override LINK library that should appear first in the STEPLIB statement.

**D** Type the data set names of the LINK libraries for the BMC products.

**E** In the Additional lib field, type the data set name for the additional LINK library that should appear last in the STEPLIB statement.
Setting the JCL options for static data sets

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Static Data Set Options panel to specify the options for sizing and cleaning up your data sets.

To set the JCL options for static data sets

1. Use the following menu selections to display the JCL Generation Static Data Set Options Update panel (Figure 70):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Static Data Set Options</td>
</tr>
</tbody>
</table>

Figure 70  JCL Generation Static Data Set Options Update panel

AJXOSDO ------- JCL GENERATION STATIC DATA SET OPTIONS UPDATE ------------------
COMMAND ==>
Type data and press Enter.

Data set sizing option... N  (N-No Sizing, B-Bmcstats, C-DB2 Catalog, O-Object Sampling)

Data set sizing device... 3390  (3380/3390)
Max cylinders ....... 99999  (Do not exceed this primary value in JCL.)
If max cylinders are exceeded, use the following for DASD data sets
Max primary quantity... 10  (Cylinders, 1 - 9999)
Max secondary quantity... 2  (Cylinders, 1 - 9999)
Max unit count.......  (Blank or 1 - 59 volumes)

Include data set cleanup step. N  (Y/N)
Return code for cleanup step .. 4  (04,08 etc.)
Temporary unit ......... SYSDA  (SYSDA, SYSALLDA, etc.)

Note See Debugging, Display and Execution Options to display sizing options in the JCL.
In the **Data set sizing option** field, type **N**, **B**, **C**, or **O** to specify the sizing method, as shown in **Table 20**:

[NOTE]

Whether or not data set sizing is performed, DB2 catalog access is required to resolve any symbolic variables. For more information about data set sizing, see “Sizing the data sets” on page 155.

---

**Table 20  Data set sizing options**

<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>no data set sizing</td>
<td><strong>N</strong></td>
<td>The product uses the default primary and secondary quantities for the permanent data sets. To modify the quantities for the primary and secondary space for a permanent data set, see step 4 on page 200.</td>
</tr>
</tbody>
</table>
| sizing by using statistics from the DASD MANAGER PLUS tables | **B** | - The DASD MANAGER PLUS product must be installed and interacting with ALTER or CHANGE MANAGER to use this option.  
- If a column is defined as a LONG VARCHAR, this option averages the row size.  
- If you select this option for Batch Execution JCL Generation, the DATASETSIZING B keyword is inserted into the AJXIN input stream.  
- Current BMCSTATS statistics should be available for objects in the worklist before you select this option. |
| sizing by using statistics from the DB2 catalog | **C** | - If a column is defined as a LONG VARCHAR, this option does not average the row size.  
- If you select this option for Batch Execution JCL Generation, the DATASETSIZING C keyword is inserted into the AJXIN input stream.  
- Current IBM RUNSTATS catalog statistics should be available for objects in the worklist before you select this option. |
| estimating sizes based on physical, random sampling of VSAM data sets | **O** | - If a column is defined as a LONG VARCHAR, this option averages the row size.  
- If you select this option for Batch Execution JCL Generation, the DATASETSIZING O keyword is inserted into the AJXIN input stream.  
- You can use this option if the statistical information in the DB2 catalog or in the DASD MANAGER PLUS tables is not current. However, JCL generation might take additional time to complete. |
You can specify whether to include comments in the Execution JCL that show statistics for determining data set sizes. For more information, see “Setting the JCL debugging, display, and Execution options” on page 203.

3 In the **Data set sizing device** field, specify the type of DASD to use in calculating the sizes of data sets.

4 In the **Max cylinders** field, specify the maximum number of cylinders for a data set.

---

**NOTE**

This option is not applicable if you dynamically allocate copy or unload data sets.

5 Specify the values that should be used for the following options when the maximum number of cylinders are exceeded:

**A** In the **Max primary quantity** field, type the number of cylinders for the maximum primary quantity.

**B** In the **Max secondary quantity** field, type the number of cylinders for the maximum secondary quantity.

**C** In the **Max unit count** field, type the maximum number of volumes.

If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options Update panels (see page 193), specify the name of a DASD unit.

6 In the **Include data set cleanup step** field, type Y or N to specify whether to generate a step in the JCL to delete the permanent work data sets.

The product deletes the data sets at the end of Execution. For Batch Execution JCL Generation, selecting this option inserts the JCLCLEANUP YES keyword into the AJXIN input stream.

You can generate the JCL for a job step that automatically deletes many of the permanent work data sets that Execution creates. Execution creates these data sets during worklist processing and sets the disposition of the data sets to new, catalog, catalog (DISP=(NEW,CATLG,CATLG)).

The cleanup job step is performed only if the condition code returned from any previous job step is less than or equal to the number that is specified in the **Return code for cleanup step** field. Table 21 lists the types of work data sets that are included in the cleanup job step. These data sets are automatically deleted unless otherwise noted.
Table 21  Work data sets in the JCL cleanup job step

<table>
<thead>
<tr>
<th>Work data set</th>
<th>ddbname</th>
<th>Used in JCL cleanup by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard (SYSDISC)</td>
<td>SYSDnnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Puncha</td>
<td>SYSPUNCH</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td>Unload (SYSREC)b</td>
<td>SYSRnnnn or</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>Rnnnyyyyyy</td>
<td>IBM REORG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The unload data sets that are used by REORG PLUS and IBM REORG are deleted automatically.</td>
</tr>
<tr>
<td>Work</td>
<td>not applicable</td>
<td>utilities that are listed in Table 22</td>
</tr>
</tbody>
</table>

a These data sets are specified in the cleanup job step but are commented out. You must edit the Execution JCL and remove the comment delimiters to delete these data sets automatically.

b Other unload data sets that are used by the BMC LOADPLUS and UNLOAD PLUS utilities and the IBM LOAD utility are specified in the cleanup job step but are commented out. You must edit the Execution JCL and remove the comment delimiters to delete the other unload data sets automatically.

Table 22 lists the work data sets that are used by the corresponding utilities.

Table 22  Work data sets used by utilities

<table>
<thead>
<tr>
<th>Work data set</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTOUT</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td>SORTPnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td>SORTOnnn</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td>SYSUTnnn</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td>SUTnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td>WRKnnn</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td></td>
<td>IBM RECOVER INDEX</td>
</tr>
<tr>
<td></td>
<td>IBM REBUILD INDEX</td>
</tr>
</tbody>
</table>
Setting the JCL options for tapes

7 In the Return code for cleanup step field, specify the value that should be compared against the condition code that is returned from any previous job step.

You can specify any two-digit value for the return code; however, BMC recommends that you specify 4.

8 In the Temporary unit field, type the name of the unit that is used to allocate temporary files when JCL is generated.

9 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.

Setting the JCL options for tapes

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Tape Options panel to specify information about tape units and stacking options.

To set the JCL options for tapes

1 Use the following menu selections to display the JCL Generation Tape Options Update panel (Figure 71):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Tape Options</td>
</tr>
</tbody>
</table>
In the **Tape Unit 1**, **Tape Unit 2**, and **Tape Unit 3** fields, type the names of valid tape units for your installation.

3 In the **Tape Volume count** field, type the maximum number of tape volumes.

4 In the **Tape EXPDT** field, type the expiration date for a tape.

5 In the **Tape RETPD** field, type the retention date for a tape.

6 In the **Tape TRTCH** field, type the parity, data conversion, translation, and compression value for 7-track tape drives as shown in Table 23.

**Table 23 Values for 7-track tape drives**

<table>
<thead>
<tr>
<th>To choose</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>not to use seven-track tape drives</td>
<td>(leave the field blank)</td>
</tr>
<tr>
<td>odd parity, conversion, and no translation</td>
<td>C</td>
</tr>
<tr>
<td>even parity, no conversion, and no translation</td>
<td>E</td>
</tr>
<tr>
<td>odd parity, no conversion, and translation</td>
<td>T</td>
</tr>
<tr>
<td>even parity, no conversion, and translation</td>
<td>ET</td>
</tr>
<tr>
<td>data compression</td>
<td>COMP</td>
</tr>
<tr>
<td>no data compression</td>
<td>NOCOMP</td>
</tr>
</tbody>
</table>
Setting the JCL options for tapes

NOTE
This option is not applicable if you choose to dynamically allocate copy or unload data sets.

7 For each type of copy or product data set, type Y or N to specify whether the data sets should be stacked on a tape with data sets of the same type.

Consider the following items if you choose tape stacking:

- Tape stacking options for baseline recovery data sets apply to CHANGE MANAGER only.
- Tape stacking options for backup SYSREC and archive data sets apply to CATALOG MANAGER only.
- Tape stacking is not applicable if you choose to dynamically allocate copy or unload data sets, unless you are using the BMC COPY PLUS utility to copy explicitly created table spaces.
- Tape stacking is not applicable if you use the Database Administration solution to execute a worklist in parallel.
- Tape stacking is disabled in the JCL for a worklist if all of the following conditions exist:
  - You use tape for unload (SYSREC) data sets, or you use tape because the maximum threshold value for DASD for a data set is exceeded.
  - You use tape stacking for unload (SYSREC) data sets.
  - You use the UNLOAD PLUS and LOADPLUS utilities and you include partitioned table spaces in the scope.

If you are using the UNLOAD PLUS and LOADPLUS utilities to multitask the unloading and loading of data, you cannot use tape stacking for unload (SYSREC) data sets. If you run the JCL, you might exceed the number of tape drives at your site.

8 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.
Setting the JCL options for temporary work data sets

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Options For Sort Files panel to specify information about the following temporary work data sets:

- SORTWORK (SORTWK)
- DATAWORK (DATAWK)
- LOGSORT (LOGSWK)

The temporary work data sets (such as SORTWK) are defined by using DISP=(,PASS) in the JCL.

To set the JCL options for temporary work data sets

1 Use the following menu selections to display the JCL Generation Options For Sort Files Update panel (Figure 72):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options Update</td>
<td>Sortwork</td>
</tr>
</tbody>
</table>

Figure 72  JCL Generation Options For Sort Files Update panel

AIXOSWO -------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE -------------------
COMMAND ===> 

Type data and press Enter.

Number of SORTWORK Data Sets... 5 (1 - 32)
SORTWORK unit name.......... SYSDA (SYSDA, 3380, etc)
Number of DATAWORK Data Sets... 5 (1 - 32)
DATAWORK unit name.......... SYSDA (SYSDA, 3380, etc)
Number of LOGSORT Data Sets... 1 (1 - 32)
LOGSORT unit name.......... SYSDA (SYSDA, 3380, etc)
Default Primary Quantity.... 10 (Cylinders)
Default Secondary Quantity... 2 (Cylinders)
SMS Data Class............... (Blank or Data Class Name)
SMS Storage Class........... (Blank or Storage Class Name)
SMS Management Class...... (Blank or Management Class Name)
$ORTPARM data set name (below)
2 Specify the options for SORTWORK data sets:

   A In the **Number of SORTWORK Data Sets** field, type the number of SORTWORK data sets.

   B In the **SORTWORK unit name** field, type the name of the unit for SORTWORK data sets.

3 Specify the options for DATAWORK data sets:

   A In the **Number of DATAWORK Data Sets** field, type the number of DATAWORK data sets.

   B In the **DATAWORK unit name** field, type the name of the unit for DATAWORK data sets.

4 Specify the options for LOGSORT data sets:

   A In the **Number of LOGSORT Data Sets** field, type the number of LOGSORT data sets.

   B In the **LOGSORT unit name** field, type the name of the unit for LOGSORT data sets.

5 If you typed N in the **Data set sizing option** field in **step 2 on page 185**, specify the default primary and secondary quantities:

   A In the **Default Primary Quantity** field, type the value for the primary quantity in cylinders.

   B In the **Default Secondary Quantity** field, type the value for the secondary quantity in cylinders.

6 Specify the IBM Storage Management Subsystem (SMS) definitions for the optional SORTOUT data set classes:

   A In the **SMS Data Class** field, type the name of the data class.

   B In the **SMS Storage Class** field, type the name of the storage class.

   C In the **SMS Management Class** field, type the name of the management class.

7 In the **SORTPARAM data set name** field, type the name of the data set that provides parameters for SyncSort.

8 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.
Setting the JCL options for permanent data sets

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the individual data set options panels to specify information about the following permanent work data sets and image copy data sets:

- Sortout (WORKDDN or LOADDN)
- Sysut (WORKDDN)
- Copy (COPYDDN, RECOVERYDDN, RECOVERDDN, ICDDN, RECOVERYICDDN, OUTCOPYDDN, or FCOPYDD)
- Sysrec (UNLDDN, INDDN, or UNLOADDDN)
- Archive (ARCHDDN)
- Cntl file (CNTLDDN)
- (CHANGE MANAGER only) Baseline recovery
- Discard (DISCARDDN)
- Error (ERRDDN)
- Map (MAPDDN)
- Report
- Punch (PUNCHDDN)
- Filter (FILTERDDN)

The permanent work data sets that contain data allow restarts. They are defined by using DISP=(NEW,CATLG) or DISP=SHR for restart or startover JCL. Examples include input (SYSUT), output (SORTOUT), discard (SYSDISC), map (SYSDC), error (SYSErr), and punch (SYSPUNCH).

Other permanent data sets are used for restart and recover purposes. They use the same dispositions as the permanent work data sets. Examples include unload (SYSREC), copy (SYSCOPY), and baseline recovery (BLRP) (for CHANGE MANAGER only).

The ROWID and LOB SYSREC data sets are used only by the UNLOAD PLUS utility and the LOB DATA MOVER program in the Database Administration solution to unload and load data contained in a ROWID column and LOB columns.
To set the JCL options for permanent data sets

1. Use the following menu selections to display the panels for permanent work data sets:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options Update</td>
<td>type of data set (SORTOUT, SYSUT, or COPY)</td>
</tr>
</tbody>
</table>

Figure 73 shows the JCL Generation Data Set Options For SORTOUT Update panel.

**NOTE**

Not all of the options are available on all of the data set options panels.

If you choose to dynamically allocate your copy or unload data sets, any changes that you make to the copy (SYSCOPY), unload (SYSREC), and baseline recovery (BLRP) (for CHANGE MANAGER only) data set options in the JCL Generation override panels do not take effect.

**Figure 73** JCL Generation Data Set Options For SORTOUT Update panel

AJXODSPL --- JCL GENERATION DATA SET OPTIONS FOR SORTOUT UPDATE ---------------
COMMAND ===> 

Type data and press Enter. Press PF3 or END to return to the main panel.

Enter Data Set Prefix below:
... APREFIX..&PREFIX..&WKID..&STEPN
Unit Name .............. SYSDA (SYSDA, TAPE, etc)
Primary Space ........ 10 (Cylinders)
Secondary Space ...... 2 (Cylinders)
Tape EXPDT. .......... (Blank or YYDDD or YYYY/DDD)
Tape RETPD. .......... (Blank or 1 - 9999 days)
SMS Data Class ....... (Blank or Data Class)
SMS Storage Class ..... (Blank or Storage Class)
SMS Management Class (Blank or Management Class)

Threshold Value ........ 0 (Cylinders, 0 means no Threshold)
Alternate Unit Name .... (SYSDA, TAPE, etc)
Alternate SMS Data Class (Blank or Data Class Name)
Alternate SMS Storage Class (Blank or Storage Class Name)
Alternate SMS Management Class (Blank or Management Class Name)
Specify the prefix for the data set.

Consider the following items when you specify the prefix:

- The suffix in the name of an unload (SYSREC) or image copy data set varies, depending on whether the data set is dynamically allocated:
  - For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.
  - For data sets that are dynamically allocated, Analysis appends a period, a two-character abbreviation, and a six-digit number to the prefix to create the name of the data set, as shown in Table 24.

### Table 24  Prefixes for dynamically-allocated data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline recovery</td>
<td>&amp;PREFIX..&amp;OBNOD.BLnnnnnn</td>
</tr>
<tr>
<td>discard</td>
<td>&amp;PREFIX..&amp;OBNOD.SDnnnnnn</td>
</tr>
<tr>
<td>local primary copy</td>
<td>&amp;PREFIX..&amp;OBNOD.LPnnnnnn</td>
</tr>
<tr>
<td>local backup copy</td>
<td>&amp;PREFIX..&amp;OBNOD.LBnnnnnn</td>
</tr>
<tr>
<td>recovery primary copy</td>
<td>&amp;PREFIX..&amp;OBNOD..P&amp;PART.RPnnnnnn</td>
</tr>
<tr>
<td>recovery backup copy</td>
<td>&amp;PREFIX..&amp;OBNOD..P&amp;PART.RBnnnnnn</td>
</tr>
<tr>
<td>primary SYSREC(^a)</td>
<td>&amp;USERID..&amp;MSSID..&amp;WORKID8.SRnnnnnn</td>
</tr>
<tr>
<td>ROWID SYSREC(^b)</td>
<td>&amp;USERID..&amp;MSSID..&amp;WORKID8.SRnnnnnn</td>
</tr>
<tr>
<td>LOB SYSREC(^c)</td>
<td>&amp;USERID..&amp;MSSID..&amp;WORKID8.SRnnnnnn</td>
</tr>
<tr>
<td>XML file reference(^d)</td>
<td>&amp;USERID..&amp;MSSID..&amp;WORKID8.SRnnnnnn.Xnn</td>
</tr>
<tr>
<td>LOB file reference(^e)</td>
<td>&amp;USERID..&amp;MSSID..&amp;WORKID8.SRnnnnnn.Lnn</td>
</tr>
</tbody>
</table>

\(^a\) If you are using the UNLOAD PLUS and LOADPLUS utilities to separate data sets for each table space partition and multitask the unloading and loading of data, Analysis also appends the P&PART variable to the prefix to create the name of the primary SYSREC data set (&USERID..&MSSID..&WORKID8.SRnnnnnn.P&PART).

\(^b\) For the ROWID SYSREC data set (ROWIDDSN), Analysis also appends .RWID to the prefix (&USERID..&MSSID..&WORKID8.SRnnnnnn.RWID). The UNLOAD PLUS utility dynamically allocates the ROWID SYSREC data set.

\(^c\) For the LOB SYSREC data set (LOBDSN), Analysis also appends .Cnnn to the prefix, where nnn is the number of the LOB column in the table (&USERID..&MSSID..&WORKID8.SRnnnnnn.Cnnn). The LOB DATA MOVER program dynamically allocates the LOB SYSREC data set.
To suppress the suffix for primary and backup copies (both local and recovery), specify Y for the appropriate POF keyword:

- FCPY_SUPPRESS_SUFF for flashcopy data sets
- PCPY1_SUPPRESS_SUFF for local primary copy data sets
- PCPY2_SUPPRESS_SUFF for local backup copy data sets
- RCPY1_SUPPRESS_SUFF for recovery primary copy data sets
- RCPY2_SUPPRESS_SUFF for recovery backup copy data sets

Note that you must ensure the uniqueness of the name of the data set.

To specify a GDG for the local and recovery image copy data sets, add the &GDG symbolic variable to the end of the data set prefix (Figure 74).
Figure 74  Using the &GDG symbolic variable

### AJXODSPL --- JCL GENERATION DATA SET OPTIONS FOR LOCAL PRIMARY COPY UPDATE ----

COMMAND ===>

Type data and press Enter. Press PF3 or END to return to the main panel.

Enter Data Set Prefix below:

- &PREFIX..&OBNOD(OGDG)

Unit Name ............... SYSDA  (SYSDA, TAPE, etc)
Primary Space ........... 10  (Cylinders)
Secondary Space ........ 2  (Cylinders)
Tape EXPDT. ............. (Blank or YYDDD or YYYY/DDD)
Tape RETPD. ............. (Blank or 1 - 9999 days)
SMS Data Class. .......... (Blank or Data Class)
SMS Storage Class ...... (Blank or Storage Class)
SMS Management Class .. (Blank or Management Class)
Threshold Value ........... 0  (Cylinders, 0 means no Threshold)
Alternate Unit Name ....... (SYSDA, TAPE, etc)
Alternate SMS Data Class ... (Blank or Data Class Name)
Alternate SMS Storage Class (Blank or Storage Class Name)
Alternate SMS Management Class (Blank or Management Class Name)

If you use the &GDG variable, JCL Generation resolves the data set name using the symbolic variable, and the name includes the GDG number (Figure 75).

Figure 75  Data set names resolved with the &GDG symbolic variable (part 1 of 2)

```c
//*--------------------------------------------------------------------
//*  UTILITY COPY DD STATEMENTS
//*--------------------------------------------------------------------

//SYCLO001 DD DSN=RDACRJ.DEMOCJ.S9(+1),
  // DCB=(SYS1.MODEL),
  // DISP=(NEW,CATLG,CATLG),
  // SPACE=(CYL,(10,2),RLSE),
  // UNIT=SYSDA
//SYCLO002 DD DSN=RDACRJ.DEMOCJ.S3(+1),
  // DCB=(SYS1.MODEL),
  // DISP=(NEW,CATLG,CATLG),
  // SPACE=(CYL,(10,2),RLSE),
  // UNIT=SYSDA
//SYCLO003 DD DSN=RDACRJ.DEMOCJ.S2(+1),
  // DCB=(SYS1.MODEL),
  // DISP=(NEW,CATLG,CATLG),
  // SPACE=(CYL,(10,2),RLSE),
  // UNIT=SYSDA
//SYCLO004 DD DSN=RDACRJ.DEMOCJ.S11(+1),
  // DCB=(SYS1.MODEL),
  // DISP=(NEW,CATLG,CATLG),
  // SPACE=(CYL,(10,2),RLSE).
```
If you specify a GDG for a dynamically allocated data set, Analysis does not append the two-character abbreviation and the six-digit number to the prefix.

**NOTE**

ALTER and CHANGE MANAGER use data set prefixes for unload (SYSREC) and image copy data sets. These prefixes ensure the uniqueness of the name of the data set when the data set is dynamically allocated. The POF that is generated in the HLQ.UDBCNTL data set includes the new prefixes, unless you copy an existing POF during installation. If you copy an existing POF during installation to the HLQ.UDBCNTL data set, you might need to modify the data set prefixes.

- The SRnnnnnn unload data set can contain the &MSSID symbolic variable, which identifies the DB2 subsystem ID (SSID) on the sending subsystem for a two-phase migration. The Analysis component also generates the -JCLP worklist command with the &MSSID variable and a value that refers to the sending SSID.

- The prefixes for copy data sets can contain the &TSSID symbolic variable, which identifies the receiving (or target) DB2 subsystem ID (SSID). If you specify the name of the DB2 SSID as the target SSID for a migrate-type work ID, the Analysis component generates a -JCLP worklist command with the TSSID.

---

Figure 75  Data set names resolved with the &GDG symbolic variable (part 2 of 2)

```
//
/* SORT WORK DD STATEMENTS */
//
/SORTWK01 DD UNIT=SYSDA,
   SPACE=(CYL,(10,2)),
   DISP=(NEW,DELETE)
/SORTWK02 DD UNIT=SYSDA,
   SPACE=(CYL,(10,2)),
   DISP=(NEW,DELETE)
/SORTWK03 DD UNIT=SYSDA,
   SPACE=(CYL,(10,2)),
   DISP=(NEW,DELETE)
/SORTWK04 DD UNIT=SYSDA,
   SPACE=(CYL,(10,2)),
   DISP=(NEW,DELETE)
/SORTWK05 DD UNIT=SYSDA,
   SPACE=(CYL,(10,2)),
   DISP=(NEW,DELETE)
/*                     */
```
Setting the JCL options for permanent data sets

parameter. The TSSID parameter specifies the name of the receiving (or target) SSID. JCL Generation uses the name of the receiving SSID to build the Execution JCL. For example, if the following command is in the worklist, JCL Generation uses the DEBA SSID for the receiving subsystem:

-JCLP 000700 MIGR TSSID DEBA

If you do not specify the name of an SSID as the target, or you specify an asterisk, JCL Generation uses the value of the &SSID symbolic variable as the value for the &TSSID variable.

**NOTE**
If the copy data set is dynamically allocated, JCL Generation substitutes the &SSID variable for the &TSSID variable in the COPY OUTPUT descriptor. The copy utility resolves the &SSID variable.

**TIP**
To specify the target SSID, see the task for defining migrate options for a migrate-type work ID in the *ALTER and CHANGE MANAGER for DB2 User Guide*.

3 In the **Unit Name** field, type the name of the unit.

Consider the following items when you specify the unit:

- If you use a tape unit, JCL Generation allocates the data sets before it opens them.

- If you have large data sets and want to avoid extents or multiple data sets on DASD, specify a tape, virtual tape, or cartridge unit.

- If you are processing a worklist in parallel, you must specify a DASD unit for your permanent work data sets.

- For the Error data set (SYSERnnn) and Map data set (SYSMAP), specify SYSDA.

If you specify TAPE and need to restart the IBM LOAD utility, you must uncatalog the existing data set. Then, you must change the DD statements to a disposition of (NEW,CATLG,CATLG).
Setting the JCL options for permanent data sets

- Tape stacking is disabled if either of the following conditions exist. As a result, specifying tape for the unit might cause you to exceed the number of tape drives at your site.

  — You use tape for the sort input (SYSUT, SUT, WRK) or sort output (SORTOUT, SORTO, SORTP) data sets, or you use tape because the maximum threshold value for DASD for a data set is exceeded. Additionally, you specify a value for the maximum number of SYSUT temporary work data sets in Analysis that the LOADPLUS and REORG PLUS utilities use to build nonclustering indexes.

  — You use the UNLOAD PLUS and LOADPLUS utilities in Analysis, and you include partitioned table spaces in the scope.

- As an alternative to specifying tape for the unit, you can specify a DASD unit that will span multiple data sets. Return to the JCL Generation Static Data Set Options panel (see page 184), and specify a value for Max cylinders and a value greater than 1 for Max unit count.

- BMC recommends that you specify SYSDA for the unit name for ROWID SYSREC unload data sets. (The ROWID SYSREC data set is used only by the UNLOAD PLUS utility and the LOB DATA MOVER program in the Database Administration solution.) If the ROWID SYSREC is set to SYSDA, the performance of the worklist commands for unloading the ROWID data set can be improved when the worklist is executed in parallel.

4 If you typed N in the Data set sizing option field in step 2 on page 185, specify the default primary and secondary quantities:

A In the Primary Space field, type the value for the primary quantity in cylinders.

B In the Secondary Space field, type the value for the secondary quantity in cylinders.

NOTE
This option is not applicable if you choose to dynamically allocate copy or unload data sets.

5 In the Tape EXPDT field, type the expiration date for a tape.

6 In the Tape RETPD field, type the retention date for a tape.
7 Specify the IBM Storage Management Subsystem (SMS) definitions for the data set classes:

A In the SMS Data Class field, type the name of the data class.

B In the SMS Storage Class field, type the name of the storage class.

C In the SMS Management Class field, type the name of the management class.

8 In the Threshold Value field, type, in cylinders, the primary quantity for the data set.

If this value is exceeded, JCL Generation uses the alternate unit and the alternate SMS parameters. Zero indicates that a threshold is not specified for the unit. If you specify zero, JCL Generation does not use an alternate unit and the alternate SMS parameters.

When ALTER or CHANGE MANAGER dynamically allocates the copy data sets for IBM COPY, it uses TEMPLATE descriptors in the worklist for local and recovery primary and backup copies. If you specify a value for Threshold Value, the product provides templates for a primary set of data set attributes and an alternate set of data set attributes. If the threshold is exceeded, the utility can perform template switching and use the alternate data set. For more information about the TEMPLATE descriptors, see Appendix B in the ALTER and CHANGE MANAGER for DB2 User Guide and the -COPY worklist command in the ALTER and CHANGE MANAGER for DB2 Reference Manual.

---

**NOTE**

If you are processing a worklist in parallel, do not specify a threshold value.

---

9 In the Alternate Unit Name field, type the alternate name of the unit to be used if the threshold value specified in step 8 is exceeded.

---

**NOTE**

The alternate unit name must be a valid tape unit name. For more information, see “Setting the JCL options for tapes” on page 188.

---

10 Specify the IBM Storage Management Subsystem (SMS) definitions for the alternate data set classes:

A In the Alternate SMS Data Class field, type the name of the data class.

B In the Alternate SMS Storage Class field, type the name of the storage class.
In the **Alternate SMS Management Class** field, type the name of the management class.

11 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.

### Setting the JCL generation data group options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Generation Data Group Options Update panel to specify information about GDGs.

### To set the JCL options for generation data groups

1 Use the following menu selections to display the JCL Generation Generation Data Group Options Update panel (Figure 76):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Generation Data Group Options (GDGs)</td>
</tr>
</tbody>
</table>

![Figure 76] JCL Generation Generation Data Group Options Update panel

AJXOGDG ---- JCL GENERATION GENERATION DATA GROUP OPTIONS UPDATE  --------------
COMMAND ===>  

Type data and press Enter.

- Define GDG base at JCL generation? . N (Y/N)
- Specify NSCR on GDG definition? . N (Y/N)
- Number of primary copy GDG entries . 10 (1-255)
- Number of recovery copy GDG entries . 10 (1-255)
- Type GDG Model data set below:
  - . SYS1.MODEL

NOTE: GDGs only apply to copy data sets.
2 In the **Define GDG base at JCL generation?** field, type **Y** or **N** to specify whether JCL Generation creates the base of the GDG.

---

**NOTE**

JCL Generation cannot create the base of the GDG if you use IBM COPY to generate image copies for implicitly created objects.

---

3 In the **Specify NSCR on GDG definition?** field, type **Y** if the base of a GDG is defined in the IDCAMS DEFINE command as EMPTY (NSCR), or **N** if the base is defined as SCRATCH (SCR):

- SCR (the default) indicates to scratch (delete) the generation data set when it is uncataloged.
- NSCR indicates to uncatalog the generation data set when the maximum number of generation data sets to keep is reached.

4 In the **Number of primary copy GDG entries** field, type the maximum number of generation data sets to keep for primary copies.

5 In the **Number of recovery copy GDG entries** field, type the maximum number of generation data sets to keep for recovery copies.

6 At **Type GDG Model data set below**, type the name of the GDG model data set.

7 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.

---

**Setting the JCL debugging, display, and Execution options**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Debugging, Display And Execution Options panel to specify information about how comments are handled in the JCL.
To set the JCL options for debugging, display, and Execution

1 Use the following menu selections to display the JCL Generation Debugging, Display And Execution Options Update panel (Figure 77):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER or CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>ALTER or CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Update Main Menu</td>
<td>Debugging, Display and Execution Options</td>
</tr>
</tbody>
</table>

![Figure 77](image)

AJXODBG JCL GENERATION DEBUGGING,DISPLAY AND EXECUTION OPTIONS UPDATE ----------
COMMAND ===>  

Type data and press Enter.

Include data set sizing comments in JCL . . . . . . N (Y/N)
Include variable substitution comments in JCL . . . . N (Y/N)
Suppress comments in JCL . . . . . . . . . . . . . N (Y/N)
NOTE: Do not set suppress comments to Y if you have specified either sizing or variable substitutions.
Specify an Alternate Program for IKJEFT01 . . . .
Post Step JCL INCLUDE member name . . . . .
Post Job JCL INCLUDE member name . . . . . .
Include in AEXIN parameters:
SYNDCDELETE . . . N (Y/N) BINDFAIL . . . . N (Y/N)
HASHFAIL . . . . N (Y/N) HASHWARNRC . . . (NUMERIC)
REBINDFAIL . . . N (Y/N) REBINDRC . . . (NUMERIC)
2MEGSQ . . . . N (Y/N) NOFAILNOIMAGECPY N (Y/N)
STOPWAIT . . . . 3 (NUMERIC)
STOPWAIT SECS . . 10 (NUMERIC)

2 Specify whether to include debugging comments in the generated JCL:

**NOTE**

BMC recommends that you include the comments if you suspect that the JCL was generated incorrectly and you need to send documentation to Customer Support. If you want to reduce the number of lines of JCL, do not include the comments.
In the Include data set sizing comments in JCL field, type Y or N to specify whether to include comments in the generated JCL that show statistics for determining data set sizes.

Comments are shown as \textit{dsso/cc}, where \textit{dsso} is the data set sizing option and \textit{cc} is a comment code. Table 25 lists the comment codes that Execution generates in the JCL.

\begin{table}[!h]
\centering
\begin{tabular}{|c|c|l|}
\hline
\textbf{Data set sizing options} & \textbf{Comment code} & \textbf{Description} \\
\hline
B, C, or O & C & uses statistics from the DB2 catalog \\
& H & uses the high relative-byte address (RBA) \\
& M & uses multiple objects to size one data set (for example, SYSUTs) \\
& N & indicates that the data set could not be sized because statistics could not be found \\
& O & uses VSAM object sampling \\
& S & uses the following formula to calculate the SORTWK size: \\
& & \( \frac{(\text{work space} \times 2)}{\text{number of SORTWK data sets}} \) \\
& W & warns that the sizing might be inaccurate \\
\hline
\end{tabular}
\caption{Comment codes for data set sizing}
\end{table}

Figure 78 shows example comments in the Execution JCL.

\begin{verbatim}
//** ERRDDN OUTPUT DD STATEMENTS
//** N = CAN'T SIZE DATA SET FOR DD SYSER001, DEFAULTS USED BECAUSE
//**  NO BMCSTATS WERE FOUND FOR TB ACMX01.T_X01PS
//SYSER001 DD DSN=RDAMCG3.MG1217D.STEP1.SYSER001, 
//         DISP=(NEW,CATLG,CATLG), 
//         SPACE=(CYL,(10,2),RLSE), ESTIMATE-B/N
//         UNIT=SYSDA 

//** SORT WORK DD STATEMENTS
//** S = SORTWK SIZE IS (WORK SPACE * 2 / #SORTWKS)
//SORTWK01 DD UNIT=SYSDA, 
//         SPACE=(CYL,(1,1)), ESTIMATE=C/S
//         DISP=(NEW,DELETE)
\end{verbatim}
Setting the JCL debugging, display, and Execution options

8 In the Include variable substitution comments in JCL field, type Y or N to specify whether to include comments that show the SLIB variables and their assigned values.

JCL Generation uses these variables to resolve the names of the data sets in the generated JCL.

3 In the Suppress comments in JCL field, type Y or N to specify whether to suppress all comments in the generated JCL.

--- NOTE ---

If you chose to include either the statistics comments or the variable comments in step 2, you cannot select to suppress all comments in the JCL.

4 In the Specify an Alternate Program for IKJEFT01 field, type the name of a program to be used instead of IKJEFT01.

--- NOTE ---

The alternate program is only used for nonworklist JCL.

5 In the Post Step JCL INCLUDE member name field, type the name of a JCL member to be included after each step in the JCL.

6 In the Post Job JCL INCLUDE member name field, type the name of a JCL member to be included at the end of a job.

7 For each AEXIN keyword, type Y or N to specify whether to include the keyword in the AEXIN input stream.

For more information about the keywords, see the list of keywords in the ALTER and CHANGE MANAGER for DB2 Reference Manual.

8 Press END to save your changes, and to return to the ALTER or CHANGE MANAGER Main Menu.
Setting the JCL utility installation options module name options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the BMC Utility Option Module Names panel to specify the name of the installation options module for the BMC utilities.

To set the JCL options for utility installation options module names

1. Use the following menu selections to display the JCL Generation BMC Utility Option Module Names Update panel (Figure 79):

   ![Figure 79 JCL Generation BMC Utility Option Module Names Update panel](image)

   From this menu | Select this item and press Enter
   --- | ---
   ALTER or CHANGE MANAGER Main Menu | Options
   ALTER or CHANGE MANAGER Main Options Menu | JCLGEN Options
   JCL Generation Update Main Menu | Utility Options
   Utility Options | BMC Utility Option Module Names

2. Specify the name of the installation options module for the BMC utilities.

   If you use any of the BMC utilities, the installation options module names will be listed in the AEXIN input stream in the JCL.
Setting the online reorg options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Online Reorg Utility Options panel to specify the options for reorganizing table spaces when using an online reorg (SHRLEVEL CHANGE) in ALTER and CHANGE MANAGER.

To set the JCL options for online reorg options

1. Use the following menu selections to display the Online Reorg Utility Options panel (Figure 80):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options panel</td>
<td>Online Reorg Options</td>
</tr>
</tbody>
</table>

Figure 80   Online Reorg Utility Options panel

NOTE: The mapping table full length cannot exceed 72 characters
2 In the **BMCREORG XBMID** field, specify the BMC EXTENDED BUFFER MANAGER (XBM) subsystem (SSID) that the BMC REORG PLUS utility accesses.

REORG PLUS uses XBM or its XBM SNAPSHOT UPGRADE FEATURE (SUF) technology to create a snapshot of the data sets to be reorganized.

The value of the SSID can be from 1 to 8 characters long.

3 In the **REORG MAPPING TABLE** field, specify the name of the mapping table that the IBM REORG utility uses to map the row IDs (RIDs) in the source table to the RIDs in the target table.

The name can be from 1 to 72 characters long.

4 Press END to save your changes, and to return to the CHANGE MANAGER Main Menu.

### Setting the Execution worklist parallelism options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Execution Worklist Parallelism Options panel to specify the options for worklist parallelism in the CHANGE MANAGER component of the Database Administration solution.

### To set the JCL options for worklist parallelism

1 Use the following menu selections to display the Execution Worklist Parallelism Options panel (Figure 81):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>ALTER and CHANGE MANAGER Options</td>
</tr>
<tr>
<td>ALTER and CHANGE MANAGER Options</td>
<td>Worklist Parallelism Options (CHANGE MANAGER only)</td>
</tr>
</tbody>
</table>
Setting the Execution worklist parallelism options

Figure 81  Execution Worklist Parallelism Options panel

<table>
<thead>
<tr>
<th>Option</th>
<th>AEXPIN keyword</th>
<th>AJXPOFIN keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Worklist in Parallel mode</td>
<td>PARALLEL YES</td>
<td>ACM_PARALLEL_WORKLST</td>
</tr>
<tr>
<td>Start XIM</td>
<td>XIMSTART YES</td>
<td>ACM_PARALLEL_XIMSTRT</td>
</tr>
<tr>
<td>Display Parallel Trace</td>
<td>TRACE YES</td>
<td>ACM_PARALLEL_XIMTRCE</td>
</tr>
<tr>
<td>Minimum XIM Initiators</td>
<td>MAXINITS</td>
<td>ACM_PARALLEL_MAXINIT</td>
</tr>
<tr>
<td>Maximum XIM Initiators</td>
<td>MININITS</td>
<td>ACM_PARALLEL_MININIT</td>
</tr>
<tr>
<td>XIM Proc Name</td>
<td>XIMPROC</td>
<td>ACM_PARALLEL_XIMPROC</td>
</tr>
<tr>
<td>XIM Group Name</td>
<td>XIMGROUP</td>
<td>ACM_PARALLEL_XIMGRP</td>
</tr>
</tbody>
</table>

JCL Generation inserts the keywords for the selected override options into the AEXPIN input stream in the JCL. These keywords override the default keywords in the POF. The Execution component uses the AEXPIN input stream to control parallel processing. Execution uses the AEXPIN input stream in conjunction with the AEXIN input stream.

Table 26 lists the options on the panel, the AEXPIN keyword, and the AJXPOFIN keyword that is overridden by the AEXPIN keyword.

2 In the Run Worklist in Parallel mode field, type Y to enable worklist parallelism.

To run a worklist in parallel, the PARALLEL keyword must also be specified in the ALUIN input stream. For more information, see the ALTER and CHÂNGE MANAGER for DB2 User Guide.
3 In the **Start XIM** field, type Y to start XIM automatically when you execute a worklist.

XIM is started only on the OS/390 or z/OS image on which you submit a job. If you use a data sharing environment, XIM is not started on other images.

4 In the **Display Parallel Trace** field, type Y to display trace messages in an output log during the execution of the worklist.

5 In the **Minimum XIM Initiators** field, type the minimum number of XIM initiators.

6 In the **Maximum XIM Initiators** field, type the maximum number of XIM initiators.

---

**NOTE**
If the maximum number of XIM initiators that you specify is larger than the number of objects in the scope of the worklist, your DASD might not be used efficiently.

7 In the **XIM Proc Name** field, type the name of the started task procedure for XIM.

---

**NOTE**
BMC recommends that the name of the started task procedure for XIM be unique for each instance of XIM that is running on an OS/390 or z/OS image. In addition, you should copy the procedure into a system PROCLIB data set. BMC also recommends that you use the suggested default value unless you have a specific requirement for another value.

8 In the **XIM Group Name** field, type the group name for XIM.

---

**NOTE**
The group name for XIM must be unique for each instance of XIM that is running on an OS/390 or z/OS image. BMC recommends that you use the suggested default value unless you have a specific requirement for another value.

9 Press END to save your changes, and to return to the CHANGE MANAGER Main Menu.

---

**Setting the worklist options**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the ALTER and CHANGE MANAGER Worklist Options panel to specify the options for sorting a worklist.
To set the JCL options for worklist sorting

1. Use the following menu selections to display the ALTER and CHANGE MANAGER Worklist Options panel (Figure 82):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>ALTER and CHANGE MANAGER Options</td>
</tr>
<tr>
<td>ALTER and CHANGE MANAGER Options</td>
<td>Worklist Options</td>
</tr>
</tbody>
</table>

Figure 82   ALTER and CHANGE MANAGER Worklist Options panel

AJXOWKL ---------- ALTER and CHANGE MANAGER WORKLIST OPTIONS ------------------
COMMAND ===> 
Type data and press Enter. Press PF3 or END to return to the main panel.
Sort Worklist . . . . . . . . (Blank - No Sorting
A - Automatic sorting
C - Object Cardinality sorting
N - Table Name sorting)
Print Sort Worklist Messages Y (Y - Print Messages,N - No messages)

2. In the Sort Worklist field, type A, C, or N (or leave the field blank) to specify the sorting method, as shown in Table 27:

<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>sorting the worklist by each table’s cardinality in a descending sequence</td>
<td>C</td>
<td>none</td>
</tr>
<tr>
<td>sorting the worklist by table order in an ascending sequence, according to the table owner and table name</td>
<td>N</td>
<td>none</td>
</tr>
</tbody>
</table>
3 In the Print Sort Worklist Messages field, type Y to record in the SYSPRINT data set and in the worklist the amount of time to sort a worklist.

4 Press END to save your changes and return to the ALTER or CHANGE MANAGER Main Menu.

### Setting user variables

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the User Defined Variables panel to specify character variables. Each variable has a corresponding symbolic variable that you can use in job cards or data set prefixes.

#### To set the JCL options for user-defined variables

1 Use the following menu selections to display the User Defined Variables panel (Figure 83):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>CHANGE MANAGER Main Options Menu</td>
<td>JCLGEN Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>User Defined Variable Values</td>
</tr>
</tbody>
</table>

---

**Table 27  Worklist sorting options (part 2 of 2)**

<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>sorting the worklist by either table cardinality or table order, depending on</td>
<td>A</td>
<td>If the Database Administration solution is used to process the worklist in parallel, Analysis sorts the worklist by table cardinality. Otherwise, it sorts the worklist by table order.</td>
</tr>
<tr>
<td>whether the worklist is processed in parallel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generating the objects in the worklist in an unsorted, random order</td>
<td></td>
<td>none</td>
</tr>
</tbody>
</table>
Figure 83  User Defined Variables panel

2 Specify the values for the variables.

The maximum length of a variable name is eight characters.

3 Press END to save your changes, to return to the CHANGE MANAGER Main Menu.

Creating a user POF

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Product Options File (POF) Functions panel to create a user POF or update the values in your ISPF profile. The panel displays the data set name of the initial POF. The panel also displays the value of the POFDATE parameter in the initial POF that was last used to update the ISPF profile.

1 Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel (Figure 84):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>Product Options File (POF) Functions</td>
</tr>
</tbody>
</table>
2 In the **Type User POF Name below** field, replace the displayed name (the initial POF) with the name of the data set for a user POF.

The name can be either an existing sequential, 80-column data set or a member of a PDS.

--- **NOTE**

Consider creating a user POF to set the options for processing a worklist in parallel.

---

3 In the **WRITE User POF data set from Profile Variables** field, type **Y** to write the ISPF variable values (located in the ISPF profile) to the user POF.
**Updating a user POF**

The product provides you two methods of updating a user POF: by using the options panels or by editing the file directly.

**To update the user POF in the JCL Generation options panels**

1. In the various options panels, specify your changes to the JCL Generation options.

2. Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE MANAGER Main Menu</td>
<td>Options</td>
</tr>
<tr>
<td>JCL Generation Main Menu</td>
<td>Product Options File (POF) Functions</td>
</tr>
</tbody>
</table>

3. In the **Type User POF Name below** field, type the name of the data set for your user POF.

4. In the **WRITE User POF data set from Profile Variables** field, type **Y**.

**To update the user POF directly**

1. From the JCL Generation Product Options File (POF) Functions panel, in the **Type User POF Name below** field, type the name of the data set for your user POF.

2. In the **BROWSE, EDIT, VALIDATE User POF** field, type **E**.

   **NOTE**

   You can edit the user POF or the initial POF by using the ISPF edit macro AJXPODAT from the **HLQ.DBCLIB** library. This library must be in your SYSPROC concatenation.

3. Edit and save the file.

4. In the **RESET All Profile Variables from User POF** field, type **Y** to update all of the ISPF variables in the ISPF profile with the variables in the user POF.
Using multiple POFs

If you want to use different values for different applications, consider using more than one user POF. When you use multiple POFs, you can reset the values from a user POF that contains specifications for a particular application. Some sample scenarios follow:

**Scenario 1**

The Payroll department needs backup copies of their data sets on tape, but other departments do not need backup copies. You can create one user POF for the Payroll department and one for the other departments. Whenever you work with payroll objects, you can specify your payroll POF name to reset the profile variables from that POF.

**Scenario 2**

Your company is a service provider for several customers. By using a separate POF for each customer, you can accommodate each customer’s naming standards.

**Scenario 3**

You have different requirements for test and production data. For test data, you want to store the copies on DASD. For production data, you want to store the copies on tape. You can use separate POFs for test data and production data.

To create an additional user POF and reset the profile variables, perform the following steps:

1. Follow the steps in “Creating a user POF” on page 214 to create an additional POF.

2. From the JCL Generation Product Options File (POF) Functions panel, in the **Type User POF Name below** field, type the name of the data set for your customized POF.

3. In the **RESET All Profile Variables from User POF** field, type Y to update all of the ISPF variables in the ISPF profile with the variables in the user POF.

Note that you would perform the same steps to specify use of a different user POF.
Refreshing the initial POF

Assume that your shop has revised its standards for naming data sets. To enforce the new standards, you need to refresh the initial POF so that users will get the updated values.

The JCL Generation component uses the value of the POFDATE keyword and the refresh attribute when determining whether to reset the ISPF profile variables to the updated POF values. The refresh attribute of a POF keyword value indicates that the ISPF profile variable should be reinitialized from the POF value if one of the following conditions exists:

- The value of the POFDATE keyword is greater than that saved in the ISPF profile.
- The name of a new initial POF is different from the name of the POF that is saved in the ISPF profile.

To refresh the initial POF, perform the following steps:

1. Edit the initial POF outside of the product.
2. Change the value of the POFDATE keyword to the current date.
3. Append the refresh attribute, (R), to the values that you want to update.
4. Save the POF.

After you save the changes, users will receive the updated ISPF variables the next time they invoke one of the products. However, the users can still use their existing user POFs, which might not contain the updated values. To ensure that the user POFs use the updated values, tell the users to reset all of their profile variables from the revised initial POF. If the users typically reset all of their variables from their user POFs, they must ensure that the updated values are included.

Alternatively, you can specify the name of a new initial POF in the POFDS installation option.
Generating POF reports

Periodically, you might need to review POF keyword values, determine whether values are missing, or diagnose a problem. The following reports can assist you:

- The **POF Validation Report** lists a POF keyword, the action taken on the keyword, and the value of the keyword in the POF. For example, you can generate this report when you create a new user POF and change the values of several keywords. The report shows the changes, the number of values that were refreshed, and any errors that resulted.

- The **Variables Initialized with Default** report lists the keywords that are missing from the initial POF and the default ISPF variables that are used to populate the keywords. You can generate this report when you want to view the new keywords and their values for a release.

**To generate the reports**

From the JCL Generation Product Options File (POF) Functions panel, in the **MSGCLASS for POF Diagnostic Messages** field, type the MSGCLASS for the SYSOUT field that is used to display messages. Consider using a SYSOUT class that is designated to go to the held queue so that you can view the output. Two SYSOUT files are allocated: AJXPOFER and AJXPOFVL.

**NOTE**

The default value for the **MSGCLASS for POF Diagnostic Messages** field is blank, which indicates that JCL Generation does not generate a report when you invoke the product.

When you invoke the product or reset the POF, the reports are listed on the output for your TSO session.

**Reusing a POF in a subsequent installation**

Assume that you customized the values in your POF, and now you are installing a new release of a product. To avoid having to customize the values again, you can specify that the Installation System use your existing POF to populate the values in the new initial POF. The new POF will contain your current values plus any new keywords (and their values) for the new release.
To reuse a POF, perform the following steps:

1 Run the Installation System.

2 From the Install System JCL Generation File Information panel, in the **Use Existing POF to Populate the New Product Options File** field, type Y (Figure 85).

**Figure 85  Reusing an existing POF**

![AJXP041 Install System JCL Generation File Information](image)

Select to execute the JCL Generation Options Dialog, which displays the initial Product Options File (POF) parameters and values. The initial POF contains information that is used by JCL Generation, a shared component of the Administrative Products. If the dialog is not executed, then a default POF will be created and placed in the product hlq.UDBCNTL data set.

- **Product Options File name** ................. AJXA1POF
- **Execute JCL Generation Options Dialog** .... N  (Y/N)
- **Use Existing POF to Populate the New Product Options File** . Y  (Y/N)

Note: Navigation and panel colors in the JCL Generation product panels are different from the Install System panels.

Press PF1 HELP for more information regarding the POF, its use, and how to avoid having to reenter the information for SSID installs.

Some of the options that you specify in the JCL Generation Options dialog will be used if you interface the Administrative Products with other BMC Software Products later in the Install System dialog.

3 Enter the names of the data set and member for the existing POF (Figure 86).

For the member name, use the name of the POF that is used as the initial POF when you invoke the product.
Overriding POF values in SLIBs

One of the primary advantages of using POFs is that you can customize your JCL without having to modify your SLIBs. Nonetheless, you might need to modify your SLIBs from time to time. SLIB variables (or ISPF variables) are used in the SLIBs. Some of these SLIB variables correspond to the parameters in the POF. Note, however, that the names of the SLIB variables differ from the names of the POF keywords.

For example, if you specify the data set prefix for local primary copies (Figure 87), the name of the primary copy data set resolves to the following name without modifications to the SLIB:

\(<\text{SSIDname}\>_.IC.T.ICPY.<\text{databaseName}\>_.<\text{tableSpaceName}\>_.<\text{ddname}\>

For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.
Now, assume that your site’s DBA decides that users should not have the ability to change the data set prefix from the options panels. The DBA can override the value in the SLIB. By specifying the value for the copy data set in the AJX#DSNS SLIB (Figure 88), the DBA can uphold your site’s naming standards.
Creating OUTPUT or TEMPLATE descriptors for copies in ALTER and CHANGE MANAGER

In ALTER and CHANGE MANAGER, you can use BMC and IBM utilities to dynamically allocate data sets that are used to create valid image copies. You can use an option in the POF to create an OUTPUT or a TEMPLATE descriptor for the Analysis component of ALTER and CHANGE MANAGER. When generating a worklist, Analysis uses that descriptor for local and recovery primary and backup copies (Figure 89).

To specify a descriptor, perform the following steps:

1. From the ALTER or CHANGE MANAGER Main Menu, select Options and press Enter.

2. Select JCLGEN Options and press Enter.

3. Select Individual Data Set Options and press Enter.

4. Select Copy and press Enter.

After changing an SLIB variable, the DBA should use JCL Generation to test the changes. If the SLIB is coded correctly, the DBA must then recompile the SLIB. The DBA can use the SLIB compiler tool that is supplied with the Administrative products to compile the SLIB. For more information about testing the changes or using the SLIB compiler, see the product documentation.

Creating OUTPUT or TEMPLATE descriptors for copies in ALTER and CHANGE MANAGER

In ALTER and CHANGE MANAGER, you can use BMC and IBM utilities to dynamically allocate data sets that are used to create valid image copies. You can use an option in the POF to create an OUTPUT or a TEMPLATE descriptor for the Analysis component of ALTER and CHANGE MANAGER. When generating a worklist, Analysis uses that descriptor for local and recovery primary and backup copies (Figure 89).

To specify a descriptor, perform the following steps:

1. From the ALTER or CHANGE MANAGER Main Menu, select Options and press Enter.

2. Select JCLGEN Options and press Enter.

3. Select Individual Data Set Options and press Enter.

4. Select Copy and press Enter.

After changing an SLIB variable, the DBA should use JCL Generation to test the changes. If the SLIB is coded correctly, the DBA must then recompile the SLIB. The DBA can use the SLIB compiler tool that is supplied with the Administrative products to compile the SLIB. For more information about testing the changes or using the SLIB compiler, see the product documentation.
5 Select Local Primary Copy and press Enter.

6 Type the name of the prefix for the data set that will be dynamically allocated (Figure 90).

Figure 90 Specifying the prefix for a dynamically allocated data set

If you are using foreground Analysis to generate the worklist, JCL Generation uses the prefix in your ISPF profile variables to create the OUTPUT descriptor. However, if you are using batch Analysis, JCL Generation generates a partial POF from your ISPF profile variables and includes this partial POF in the batch Analysis JCL (Figure 91).

Figure 91 Excerpt of batch Analysis JCL with POF keywords

```plaintext
//AJXPOFIN DD * 

PCPY1_DATACLASS_ALT=
PCPY1_DATACLASS=
PCPY1_EXPDT=
PCPY1_MGMTCLASS_ALT=
PCPY1_MGMTCLASS=
PCPY1_PREFIX=PAYROLL.IC.LP..&DB..&TS
PCPY1_PRIQTY=10
PCPY1_RETPD=
PCPY1_SECQTY=2
PCPY1_STACK=N
PCPY1_STORCLASS_ALT=
```
Where to go from here

Now that you have set up ALTER or CHANGE MANAGER for your environment, you can accomplish the various goals that are listed in the following table.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrate database environments</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide, Chapter 1</td>
</tr>
<tr>
<td>take a snapshot of a database environment</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide, Chapter 2</td>
</tr>
<tr>
<td>compare database environments</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide, Chapter 3</td>
</tr>
<tr>
<td>create or alter objects in a database environment</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide, Chapter 4</td>
</tr>
<tr>
<td>maintain database environments</td>
<td>ALTER and CHANGE MANAGER for DB2 User Guide, Chapter 5</td>
</tr>
</tbody>
</table>
Glossary

A

ACM
The product code that BMC uses to identify the CHANGE MANAGER product.

ACT
The product code that BMC uses to identify the CATALOG MANAGER product.

action code
A one-character or two-character command that you can enter on one of the lines of a list panel. Although you can only enter one command per line, you can enter multiple commands on a single panel.

Administrative Products for DB2
A collection of products from BMC that includes ALTER, CATALOG MANAGER, CHANGE MANAGER, and DASD MANAGER PLUS. These integrated products are designed to help database administrators, system programmers, and application developers automate the tasks associated with the implementation and administration of a DB2 Universal Database for z/OS system.

When all Administrative products are installed properly, they can access some of each other’s functionality. In addition, these products can access the IBM and BMC utilities.

AEXIN
The ddname of the input stream that the Execution component uses.

AEXPIN
The ddname of the input stream that the Execution component uses to control parallelism processing.

AEXPRINT
The ddname of diagnostic output of the Execution component. This diagnostic output data set contains all output from the Execution process, including DB2 for z/OS utility messages, BMC utility messages, dynamic SQL messages, IDCAMS messages, and any other messages that are generated by the actions of the worklist. AEXPRINT is frequently referred to as the worklist execution log.
AEXPRnnn
The ddname of one of the initiator outputs of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains AEXPRINT output from each BMC Cross-System Image Manager (XIM) initiator.

AEXPTRAC
The ddname of the trace output of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains tracing records.

AEXSYnnn
The ddname of one of the initiator outputs of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains system messages and job information for each BMC Cross-System Image Manager (XIM) initiator.

AJX variables
A group of variables (JOB, STEP, and DD) that are used during Job Control Language (JCL) generation. JOB global variables are set once per JCL creation session. STEP global variables are set at the beginning of a JCL creation session and are updated when a new JOBSTEP is detected. Data Definitions (DD) variables are local variables that are set for each creation of a JCL DD entry. Descriptions of these variables are provided in HLQ.DBSLIB($AJXDOCV).

AJXIN
The ddname of the input stream that the Batch Execution JCL Generation component uses.

AJXPOFIN
The ddname of the override input stream that the Batch Execution JCL Generation component uses.

AJXPOFVL
The ddname of the product options file (POF) validation report that the Batch Execution JCL Generation component uses.

AJXPRINT
The ddname of the diagnostic output of the Batch Execution JCL Generation component.

alloc unit
The allocation unit that is used for space estimation calculations, volume placement, and primary and secondary quantities. Possible values are K (kilobytes), T (tracks), or C (cylinders). The default value comes from the installation options modules.

ALTER for DB2
A BMC product that provides advanced database administration and manipulation within a single DB2 subsystem. ALTER streamlines the process of changing and migrating database objects, handles analysis for both changes and migrations, and automatically generates SQL, DB2 commands, and utilities.
alter-type work ID
A work ID that ALTER and CHANGE MANAGER use to perform modifications to the local DB2 subsystem.

ALU
The product code that BMC uses to identify the ALTER product.

ALUIN
The ddname of the input stream that the Import, Baseline, Baseline Report, Compare, Analysis, and CM/PILOT components use.

ALUPRINT
The ddname of the diagnostic output of the Import, Baseline, Baseline Report, Compare, and Analysis components.

Analysis
A component of ALTER and CHANGE MANAGER that reads the change and migration requests for a particular work ID and generates a worklist to implement the requests. Analysis checks for consistency with the DB2 catalog before it builds the worklist.

Application
In the CM/PILOT component, the association of a group of CHANGE MANAGER profiles that are used repeatedly for the change management tasks of a specific DB2 application.

ASU
The product code that BMC uses to identify the DASD MANAGER PLUS product.

attribute
A value that defines certain properties of an object. Each attribute of an object can occur only once and has a single value from a finite list of possible values. For example, some of the attributes of the TABLE object are database, table space, and EDITPROC. Attributes differ from sub-elements because sub-elements can occur multiple times and can have attributes of their own. For example, columns are sub-elements of tables.

AUTHID
See authorization ID.

authorization ID
An identifier that is allowed a set of privileges. An example of the authorization ID is the owner of a table space, database, storage group, or synonym. An authorization ID is the implicit qualifier of a table, view, alias, or index name.

auxiliary index
An index on an auxiliary table. Each index refers to a large object (LOB) column.
auxiliary list
A list of several auxiliary objects (such as auxiliary table spaces, tables, or indexes) of only one type.

auxiliary table
A table that contains a single large object (LOB) column. An auxiliary table resides in an auxiliary (or LOB) table space.

auxiliary table space
A nonpartitioned table space that contains the data for a large object (LOB) column in an auxiliary table.

B

base table
A table that contains a ROWID column and the definition for a large object (LOB) column. A base table is incomplete if it does not contain any auxiliary objects, such as auxiliary table spaces, tables, and indexes.

base table space
A table space that contains base tables.

baseline
A set of data structures that are captured at a specific point in time.

Baseline
A component of CHANGE MANAGER that captures a set of DB2 structure definitions from either the DB2 catalog or a DDL file at a specific point in time.

baseline name template
A template that a baseline profile can contain. This template is used to create the names of the baselines that are established with the profile. The template can include arbitrary text and the special character sequences #### and @@@@@@ or @@@@@@@@, or a combination of both types of sequences. When the baseline is created, ascending numbers replace the #### sequence and the current date replaces the @@@@@@ or @@@@@@@@ sequence. For @@@@@@, the date is in the form YYMMDD. For @@@@@@@@, the date is in the form YYYYMMDD.

baseline profile
A BMC object that contains the information that is necessary to establish a baseline. The two types of baseline profiles are catalog and DDL.

batch component
A component that can run in batch mode. Batch components include Analysis, Baseline, Baseline Report, Compare, Execution, and Import.
BMC object
A logical entity that contains data that is necessary for performing tasks. BMC objects have a two-part name \((a.b)\). The BMC objects include work IDs, worklists, unload data sets, baselines, profiles, CDL files, DDL files, internal tables, task IDs, CM/PILOT worklists, applications, and script tables.

BMCCOPY
The BMC COPY PLUS utility that is used to create an image copy. The short form of the command is BMCI.

BMCLOAD
The BMC LOADPLUS utility that is used to load DB2 tables. The short form of the command is BMCL.

BMCREORG
The BMC REORG PLUS utility that is used to perform a reorganization. The short form of the command is BMCR.

BMCSTATS
A command that invokes the statistics collecting function of the DASD MANAGER PLUS product. The short form of the command is BMCS. BMCSTATS is similar to the IBM RUNSTATS utility. It provides the same statistics plus many additional statistics.

BMCUNLOAD
The BMC UNLOAD PLUS utility that is used to unload data from a full image copy of one or more tables in a table space. The short form of the command is BMCD.

C
catalog baseline
A baseline that is established on part of the DB2 catalog. A catalog baseline must include a scope, which might be defined directly in a baseline profile or by reference to a scope in a migrate profile.

catalog indirection
An optional method of implementing the Administrative products that allows them to access the DB2 catalog indirectly when making information queries. Catalog indirection is accomplished by using synonyms that point either to a copy of the DB2 catalog or to user-created views of the catalog. Benefits include reducing catalog contention and providing an additional level of security for sensitive catalog data.

CATALOG MANAGER for DB2
A tool that is designed to automate the day-to-day tasks associated with administering DB2. This product provides an interactive, intuitive, and easy-to-use interface for submitting DB2 commands and retrieving catalog information using qualified lists, wildcard searches, and dependency lists. CATALOG MANAGER provides the ability to create and drop DB2 objects, re-create dropped structures and data, and browse and edit table data.
CDL
See Change Definition Language.

Change Definition Language (CDL)
A BMC proprietary language that is used to specify changes to DB2 data structures.

CHANGE MANAGER for DB2
A BMC product that enables database administrators, system administrators, and developers to manage user applications and individual database objects globally. As a robust extension of the ALTER product, CHANGE MANAGER automates data structure changes across multiple DB2 subsystems and between DB2 and data modeling tools by providing a way to implement, migrate, synchronize, and back out data structure changes while preserving structure modifications that might be unique to a specific subsystem.

change rule
An element of a migrate-type work ID or a migrate profile. Change rules define the automatic object attribute changes that are made during a migration or change migration process. These rules can include changes, or sub-element inclusion or exclusion. Change rules can also force VCAT definitions for table spaces and indexes.

checkpoint
A point at which information about the status of a job and the system can be recorded so that the job step can later be restarted.

See also sync point.

child
A DB2 object that contains the foreign keys which reference the primary key in a parent table.

See also parent.

CLIST
Command list.

CM/PILOT for DB2
A component of CHANGE MANAGER that automates the DB2 change management processes that you perform using CHANGE MANAGER.

CM/PILOT worklist
A data set that contains the ordered commands, keywords, and parameters that CHANGE MANAGER needs to process a task ID.

command
A token that you can enter at the command prompt on a panel.

See also action code.
commit
   An operation that terminates a unit of work. A commit releases all locks. Data that was changed is now consistent.

Compare
   A component of CHANGE MANAGER that identifies the differences between two sets of data structures and then generates a CDL file. You can review the differences and decide how to apply the differences to implement the necessary changes. You can compare data structures that are stored in a DDL file, baseline, worklist, or DB2 catalog.

Compare1
   The primary input to the compare process.

   See also source.

Compare2
   The secondary input to the compare process.

   See also target.

component
   A major functional unit of ALTER or CHANGE MANAGER, such as Analysis, Execution, Specification, or Import.

constraint
   See referential constraint.

Cross-System Image Manager (XIM)
   A BMC technology that provides sysplex performance improvements by enabling the distribution and management of discrete units of work (UOW) across one or more OS/390 and z/OS systems. The BMC products that use XIM can divide single, long-running tasks into multiple parallel tasks that can be run across multiple computers in the sysplex, thus decreasing the overall elapsed time. The products can also be used with XIM in a data sharing environment on a single OS/390 or z/OS image.

D

DASD MANAGER PLUS for DB2
   A BMC product that automates utility generation, gathers comprehensive statistics, monitors changes in the database, and enables you to perform maintenance based on the condition of the data instead of a rigid schedule.

Data Control Language (DCL)
   A category of SQL statements that control data security.

data definition language (DDL)
   A category of SQL statements that create, modify, or delete database objects.
data definition name (ddname)
   The name of a data definition (DD) statement in job control language (JCL) that corresponds to a data control block that contains the same name.

Data Manipulation Language (DML)
   In the CM/PILOT component of CHANGE MANAGER, SQL-like statements that can be used to update, delete, and migrate data structures.

data set sizing
   The process of determining data set allocations, especially as used by the JCL Generation component. Data set sizing is distinct from space estimation.

data structure
   An object that is defined in the DB2 catalog. Objects include storage groups, databases, table spaces, tables, indexes, foreign keys, views, synonyms, aliases, and triggers.

database administrator (DBA)
   An individual who is responsible for the design, development, operation, security, maintenance, and use of databases.

database request module (DBRM)
   A module that contains SQL statements which the DB2 precompiler has extracted from a source program.

DBA
   See database administrator.

DB2 catalog
   System tables, maintained and used by DB2, that contain descriptions of DB2 objects such as tables, views, and indexes.

DB2 command
   An instruction to the DB2 subsystem. Some example command processes enable you to start or stop DB2, display information on current users, start or stop databases, and display information about databases. DB2 commands always begin with a hyphen (–).

DBCS
   See double-byte character set.

DD statement
   Data Definition statement.

DDL
   See data definition language.

DDL baseline
   A baseline that is established on a file that contains DDL.
ddname
   See data definition name.

default options module
   See installation options module.

default value
   A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

delimited identifier
   An SQL identifier that is enclosed within escape characters.

dependencies
   The name or values of objects which another object uses as part of its definition or as a hierarchical subordinate.

dependent
   A child object (row or table) that has at least one parent.

dependent object
   An object whose definition relies on the name or the values of another object. The dependent object references the other object.

destination
   The intended receiving location for CDL or a worklist.

device type
   The type of disk device used for DB2 data set allocation, such as 3380 and 3390, or generic types TAPE and CART.

double-byte character set (DBCS)
   A delimited set of characters in which each character is represented by two bytes. Katakana and other lowercase characters are nonstandard characters and must be contained within double quotes.

DOPTS
   See installation options module.

duplicate
   An action that you can make on a database object. You can duplicate single or multiple objects within the same database system. If you want to create an object that is like an existing object, you can use the Create Like command to duplicate the existing object. You can then change the object name and make any other necessary changes.
**E**

**edit procedure**

See EDITPROC.

**EDITPROC**

An edit procedure that defines an editing routine to be invoked just after a record that corresponds to a table row is retrieved and just before that record is stored. Editing routines allow for data compression, decompression, and encrypting.

**END**

An ISPF command, similar to **Enter**, that validates and processes the information on a panel and returns you to the previous panel, but does not execute commands. This command is typically programmed on your keyboard as a function key, such as **PF3**.

**Enter key**

The key that executes any commands that have been specified. For a sequence of panels, the **Enter** key displays the next panel.

**Execution**

A component of ALTER and CHANGE MANAGER that carries out the commands in a worklist.

**exit routine**

A program (BMC, IBM, or user-written) that receives control from DB2 to perform specific functions. Exit routines run as extensions of DB2 (for example, authorization checking).

**F**

**field procedure**

See FIELDPROC.

**FIELDPROC**

A user-written exit routine that is designed to receive a single value and transform (encode or decode) it in any way that the user specifies.

**foreground component**

A component that can run in foreground mode. Foreground components include Front End, Specification, Analysis, Compare, JCL Generation, and Import.

**Front End**

A component of ALTER and CHANGE MANAGER that acts as the interface between the user and the other components. Front End is an interactive ISPF dialog that is responsible for creating and maintaining BMC objects and facilitating the generation of JCL.

**full-recovery baseline**

A baseline that captures data and the data structure definitions at a specific point in time.
GDG

See generation data group.

generation data group (GDG)
A finite number of data sets that are kept in chronological order. Each data set is a generation data set.

Group ID
In an outbound migrate profile, a four-character identifier that links locations (or application instances) together. If a migrate profile defines one or more Group IDs, Analysis and Compare generate a single output file (worklist or CDL) for each group. If groups are not defined, Analysis and Compare generate one output file for each location. Locations within a group must reside on the same physical DB2 subsystem.

hash value
A number that appears at the end of commands in worklists or CDL files. The hash value is generated based on the contents of the command line and allows the products to determine whether the line has been manually modified since the file was generated.

HLQ
High-level qualifier of a data set.

image copy
An exact reproduction of all or part of the data in a table space. IBM provides utility programs to make full image copies (copy the entire table space) or incremental image copies (copy only the pages that have been modified since the last image copy). The BMC COPY PLUS utility can perform the same function. You can make an image copy of an index.

import
The process of obtaining an object or an object-set definition from a file or an external database and applying it to an alter-type work ID.

Import
A component of CHANGE MANAGER that converts statements that are stored in a CDL or DDL file into change requests in a work ID. In ALTER, Import converts statements that are stored in a DDL file.

inbound migrate profile
A profile that can be used with the Import component to automate changes to object attributes. An inbound migrate profile can contain only change rules (no locations or scope).
incremental DDL
The DDL that changes the data structures that exist in the DB2 catalog, by using either an alter strategy or a drop-then-rebuild strategy.

installation options module
An assembler module that contains keywords whose global values determine the operating environment for a BMC product.

Interactive System Productivity Facility (ISPF)
An IBM product that provides a full-screen editor and dialog manager.

internal table
A table that ALTER or CHANGE MANAGER use to store information.

ISPF
See Interactive System Productivity Facility.

ISPF skeletons
Data definition statement templates that JCL Generation uses. The skeletons are described in HLQ.DBSLIB($AJXDOC).

J

JCL
Job Control Language that is used to execute processes in batch mode.

JCL DSN
The name of the data set that contains job control language (JCL). The data set must exist and can be partitioned or sequential. You must specify a member name for partitioned data sets. You can use symbolic variables.

JCL Generation (JCLGEN)
A component of ALTER and CHANGE MANAGER that constructs a job control language (JCL) file for running the components in batch mode. When you choose to build JCL, JCLGEN is passed to the worklist that contains the control statements. ALTER and CHANGE MANAGER resolve all data set names that are entered with symbolic variables on the interface panels. JCLGEN resolves all data sets that are passed from the option panels and the unload data sets that are used by the Execution facility. The generated JCL includes data definition statements (ddnames) for all data sets that are needed by Execution, as well as the EXEC statement for the program and any necessary control parameters.

JCL variable display
The resulting output of a user option that includes debugging comments within any generated JCL. All AJX-prefixed variables are displayed as //* comments in the JCL to assist in diagnosing JCL Generation problems.
JCLGEN
See JCL Generation.

job
A batch unit of work that is defined by JCL, a work ID, and a worklist to perform tasks. ALTER and CHANGE MANAGER use the information that you supply for a work ID to generate the worklist, which provides the BMC JCL Generation component (JCLGEN) with the information that is necessary to build the JCL to run a job.

large object (LOB) column
A type of column that is used to store large objects (LOBs), such as images, audio, video, text, or graphics, as strings. The data type of the column is defined as LOB (such as a binary large object, or BLOB; character large object, or CLOB; or double-byte character large object, or DBCLOB) or as a distinct or user-defined type (UDT) that is based on a LOB data type. A LOB column resides in an auxiliary table.

large object (LOB) table space
See auxiliary table space.

LINK library
A partitioned, cataloged data set that is used to store and retrieve all or part of a program in a form that is suitable for loading into main storage for execution. The LINK library contains executable modules that perform a product’s processes.

LOAD
A command that invokes the IBM LOAD utility. The short form of the command is LOAD. The utility loads data from a file to a table space.

location
An arbitrary identifier that groups change rules for a particular destination. An outbound migrate profile can define one or more locations. A location is not a DB2 location identifier.

See also application instance.

LOB DATA MOVER
In the Database Administration solution, a program that is used to unload and load data that is contained in large object (LOB) columns.

menu
A list of action options. You select an action by typing its corresponding number in the option input field and pressing Enter. A menu panel might contain other fields that you can use to qualify the action.
**migrate**
- The process of moving DB2 data structures, data structures and data, or data structure changes, from an origin to a destination subsystem.

**migrate profile**
- A profile that can be reused to select a set of objects, to customize changes to objects which are migrated to different locations or used in a comparison, or both.

**migrate-type work ID**
- A work ID that is used to migrate data structures, data, or both. Worklists that are generated from a migrate-type work ID do not contain SQL DROP statements. These work IDs can have migrate options and change rules.

**mixed list**
- A panel in the Specification component that can display multiple DB2 object types.

**N**

**name propagation**
- The process of extending to dependent objects the changes that you make in a referenced object. For example, if you change the name of a table that is referenced in a view, name propagation replicates the new name in the view definition.

**name template**
- See baseline name template and work ID name template.

**null**
- A special value that indicates the absence of information.

**O**

**object**
- A DB2 entity that can be created or dropped. Objects are storage groups, databases, table spaces, tables, indexes, foreign keys, views, synonyms, aliases, or triggers. Unlike DB2, which treats foreign keys and check constraints as table attributes, ALTER and CHANGE MANAGER treat foreign keys and check constraints as independent objects.

**object list**
- A list of one or more objects of different types.

**option**
- A named value that is used to control one or more components. Global options are defined in the installation options module. The user can override the installation options by specifying the user options or by specifying keywords in the component’s ALUIN, AEXIN, or AJXIN input stream.
origin
   The sending location or source of a migration or a change migration.

orphaned auxiliary index
   An auxiliary index that is not associated with an existing auxiliary table.

orphaned auxiliary table space
   An auxiliary table space that does not contain an auxiliary table.

outbound migrate profile
   A profile that is used in performing a change migration. An outbound migrate profile might
define one or more locations that contain change rules that are defined for any or all of those
locations. Optionally, an outbound migrate profile can have a scope to select the DB2 objects on
which to operate.

P

page
   A unit of storage within a table space (4K or 32K) or index space (4K). A page in a table space
contains one or more rows of a table. 8K and 16K pages can be used.

parent
   A DB2 object that contains the primary key which might be referenced by one or more foreign
keys in the child table.

   See also child.

partitioned data set (PDS)
   A data set in direct access storage that is divided into partitions, called members, each of which
can contain a program, part of a program, or data. Synonymous with program library.

partitioned table space
   A table space that is subdivided into parts (based upon index key range), each of which can be
processed independently by utilities.

pattern
   A rule that is applied to the naming of objects of a specified type. You can use wildcard
characters (\% and \*) when you define a pattern.

PDS
   See partitioned data set.

piecesize
   For a nonpartitioned Type 2 index, the maximum size of a data set storage piece in kilobytes.
Valid sizes are 0, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288,
1048576, 2097152, and 4194304.
privilege
The capability of performing a specific function (authorization) on an object. Privileges might be explicitly or implicitly granted.

profile
A collection of scope rules, change rules, and locations that enables you to define and control a change, migrate, or baseline process.

See also baseline profile, migrate profile, inbound migrate profile, and outbound migrate profile.

protected baseline
A baseline that is designated as protected from deletion. A protected baseline cannot be deleted until the protected designation is removed.

See also unprotected baseline.

R
receive-type work ID
A work ID that is used to create new data structures and load migrated data on a different subsystem.

recovery
The process of restoring a set of data structure definitions to their state at a particular point in time. Recovery involves comparing the DB2 catalog to a baseline, importing the CDL, generating a worklist with Analysis, and executing the worklist. If the baseline is a full-recovery baseline, you can recover data and the data structure definitions.

reference location
A location in a profile whose rules are used by another location. For example, a location called Houston might have that rules that are explicitly defined, while locations called Austin and Dallas might use Houston as a reference location.

reference profile
A profile whose scope is used by another profile. A baseline profile can reference a migrate profile, and a migrate profile can reference a baseline profile. Using a reference profile enables you to define the scope only once and thus eliminates errors that might arise from redesigning the scope.

referenced object
An object that a dependent object references. If you change the definition of a referenced object, dependent objects might not continue to function properly.

referential constraint
The requirement that nonnull values of a designated foreign key are valid only if they equal values of the primary key of a designated table. The relationship between the primary key in the parent table and a foreign key in a dependent table is used to establish referential integrity in a
database. A referential constraint is always assigned a name to distinguish it from other constraints.

referential integrity
The condition that exists when all intended references from data in one column of a table to data in another column of the same or a different table are valid. Maintaining referential integrity requires enforcing referential constraints on all LOAD, RECOVER, INSERT, UPDATE, and DELETE operations.

REORG
A command that invokes the IBM REORG utility. The short form of the command is REOR.

S
SBCS
See single-byte character set.

scope
The final set of DB2 objects that are selected from the catalog on which the product operates. A scope includes objects that are explicitly selected and any of their object dependencies. For the Baseline component, the scope selects the objects that are captured in the baseline. For the Compare and Analysis components, the scope selects the objects that are included in the comparison or analysis process. A scope consists of one or more scope rules.

scope rule
A specification for selecting DB2 objects from the catalog by object type and name.

Script table
In the CM/PILOT component of CHANGE MANAGER, ordered steps that prompt you for the information that is required to perform a change management task.

selection list
A list of related items from which you can select one for further action. The actions (line commands) that you can specify in the Act field are typically displayed across the top of the panel.

single-byte character set (SBCS)
A character set in which each character is represented by a one-byte code.

source
In CHANGE MANAGER, the original object of a migration or the original object of a Compare process. When migrating objects or databases, the source database is the database from which you are migrating. The Compare process compares a source data structure with a target data structure. This process synchronizes two data structures, and the source is the data structure that needs to change.

See also Compare1.
Space Estimation
A feature of ALTER and CHANGE MANAGER that enables you to determine the amount of space that a table space or index will require, based on the object definitions and their estimated usages.

Specification
A component of ALTER and CHANGE MANAGER that enables you to create or edit data structure change or migration requests. Specification stores its change or migration requests in a work ID.

SQL
See Structured Query Language.

SQLID
The authorization ID that is used as the implicit qualifier of table, view, synonym, and index names in dynamic SQL statements. The SQL ID, along with the other authorization IDs of a process, is used for authorization checking of dynamic SQL statements.

SSID
A DB2 subsystem identifier.

sequence number
A six-digit, zero-filled number that identifies a statement in a worklist. The sequence number appears in columns 7 through 12 of the first line of each worklist command.

Structured Query Language (SQL)
An ANSI-standard language for database definition, manipulation, and query.

structure-only baseline
A baseline that contains only data structure definitions. No data from those data structures are included.

sub-element
A component of an object. For example, a column is a sub-element of a table, and a volume is a sub-element of a storage group.

subtype
See data subtype.

symbolic variable
A user interface variable that has its value set interactively for the current user and session at the time of job control language (JCL) generation. The BMC JCL Generation component (JCLGEN) uses symbolic variables to perform ISPF file tailoring services. A symbolic variable should be preceded with an ampersand (&). In the installation options modules, a symbolic variable
should be preceded with two ampersands (&&). Symbolic variables should not be confused with global job variables (AJX-type), which have their values set for all users and all sessions.

*See also* JCL Generation.

**SYNC**

A worklist command that invokes a checkpoint to use for restart processing and commits data to DB2.

**sync point**

A completion flag that is set during the execution of a worklist. The Execution program writes sync points to the SYNC table whenever it encounters –SYNC or –STOP commands in the worklist input stream. All SQL statements between sync points are executed as a single DB2 transaction. If a worklist is halted before completion for any reason, sync points enable you to begin processing the worklist from the last sync point.

**synchronization**

The process of identifying structural differences between two copies of the same data structure and then making the data structures identical. For example, separate groups of developers might be independently modifying several copies of a set of data structures. At various times, the copies need to be synchronized to ensure that all of the developers are using the same structure definitions.

**T**

**table space**

A page set that is used to store the rows of one or more tables. A table space represents one or more physical data sets.

**target**

The object of a migration or the object of a Compare process. When migrating objects or databases, the target database is the database to which you are migrating. The Compare process compares a source data structure with a target data structure. This process synchronizes two data structures, and the target is the data structure that contains the wanted changes.

*See also* Compare2.

**task ID**

A unit of work in the CM/PILOT component.

**template**

A method of obtaining specifications for the definition of an auxiliary table space, table, or index. This definition can be replicated for each of the partitions in the base table space.

*See also* baseline name template and work ID name template.
unit
A specific device, device type, or group of devices that are used in data set allocation.

unload data set
An object that is used to store data while DB2 objects are dropped and rebuilt. The unload data set is also referred to as a SYSREC data set.

UOW
Unit of Work. A unit of work consists of the worklist commands that are bounded by the -BEGU and -ENDU commands and that are run in a BMC Cross-System Image Manager (XIM) initiator.

UOWTRnnn
The ddname of the unit of work (UOW) output of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains tracing records for each BMC Cross-System Image Manager (XIM) initiator.

unprotected baseline
A baseline that is not designated as protected from deletion.

See also protected baseline.

user options
A set of options that are stored in the user’s profile and that are used by Front End and JCL Generation for running ALTER or CHANGE MANAGER components. The user options are initially set from the installation options module the first time that the user runs ALTER or CHANGE MANAGER.

V
variable
See symbolic variable.

VCAT allocation
A volume placement parameter that specifies the data set high-level qualifier that is appropriate for the DB2 subsystem. Use this field for nonstorage group allocation only.

versioning
The process of comparing baselines that represent the same set of structures at different points in time in order to change a version of a data structure.
**W**

**wildcard**
A symbol that you can use to represent a value in SQL statements, filters, and name patterns. Valid wildcards for SQL statements and filters include the following symbols:

- the % and * represent any character string
- the _ and ? represent a single character

**work ID**
A unit of work with a two-part name (`owner.name`) that contains change or migration requests in change definition (CD) tables. The change or migration requests can be either imported or created manually through the Specification component.

**work ID name**
A work ID name is a string of up to eighteen alphanumeric characters excluding percent (%), asterisk (*), underscore (_), and space. When you specify a work ID name, you can use a wildcard pattern to display a group of similar names.

**work ID name template**
A template from which the name of a work ID is created. You can use this template to create or replace a work ID in batch mode. The name template contains the special character sequences #### and @@@@@@@ or @@@@@@@@@, in addition to text characters. An ascending sequence of numbers replaces the #### sequence, and the current date replaces the @@@@@@ or @@@@@@@@@ sequence when the work ID is generated. You can create and replace receive-type work IDs (using name templates) with the Import and Execution components.

**work ID owner**
The Authorization ID of the creator of the work ID.

**worklist**
A data set that contains commands for implementing a data structure change or migration.

**worklist execution log**
*See AEXPRINT.*

**worklist parallelism**
A feature in the Database Administration solution that reduces the elapsed time for executing a worklist that is generated by the CHANGE MANAGER product.

**X**

**XIM**
*See Cross-System Image Manager.*

**XIM initiator**
A program that executes one or more units of work (UOW).
Index

Symbols
$AJXDOC data set 154
% (percent sign) as a wildcard 142
* (asterisk) as a wildcard 142
? (question mark) as a wildcard 142
_ (underscore) as a literal 143
as a wildcard 142

A
above-the-bar storage 154
access
controlling 145
to database 160
to parallelism feature, restricting 152
ACM_PARALLEL_MAXINIT AJXPOFIN keyword 210
ACM_PARALLEL_MININIT AJXPOFIN keyword 210
ACM_PARALLEL_WORKLST AJXPOFIN keyword 210
ACM_PARALLEL_XIMGRP AJXPOFIN keyword 210
ACM_PARALLEL_XIMPROC AJXPOFIN keyword 210
ACM_PARALLEL_XIMTRCE AJXPOFIN keyword 210
ACMvryA Analysis plan 146
ACMvryB Baseline plan 147
ACMvryC Compare plan 147
ACMvryE Environment plan 146
ACMvryF Front End plan 146
ACMvryI Import plan 146
ACMvryL LOB Data Mover plan 147
ACMvryR Report plan 147
ACMvryS Specification plan 147
Administrative Assistant 50
AEXAUNLD package 148
AEXEFAUH bit 151
AEXESTDL package 148
AEXIN input stream, user options 136
AEXPIN input stream
user options 136
using in a parallel worklist 210
AEXSQLIO package 148
AEXUSERX DSECT 149
AEXvrHA Execution Monitor Entry plan 148
AEXvrHM Execution Monitor plan 148
AJXIN input stream 136
AJXPODAT edit macro 216
AJXPOFER message file 219
AJXPOFIN, user options 136
AJXPOFVL message file 219
aliases, INSERT statement 162
allocating data sets 141
ALTER
architecture 30
component of Administrative Assistant 50
components 32
Execution plans 147
objects 35
overview 22
plans, list of 146
ALTER objects
DDL file 37
internal table 37
unload data set 37
work ID 35
worklist 36
alter-type work ID 35
ALUDSYNC member 159
ALUEUSX1 member 149
ALUIN input stream, user options 136
Analysis component
default data set name 168
of ALTER 33
of CHANGE MANAGER 40
setting default processing options 167
Application ID 138
applications, user options 178
asterisk (*) as a wildcard 142
ASXEFLAG 151
authorization switching, controlling 149
B
Baseline component
default data set name 174
of CHANGE MANAGER 40
setting default processing options 175
Baseline Report component
default data set name 176
setting default processing options 176
baselines
default run type 175
deleting 159
object 44
profile 45
table maintenance 159
table sizing and growth 159
worklist 44
BL_ALIAS product table 157
BL_AUXTABLE product table 157
BL_CHECK product table 157
BL_CHECK2 product table 157
BL_COLAUTH product table 157
BL COLUMN product table 157
BL_DATABASE product table 157
BL_DBAUTH product table 157
BL_INDEX product table 157
BL_KEY product table 157
BL_KEYTARGETS product table 157
BL_REL product table 157
BL_RELKEY product table 157
BL_RESAUTH product table 157
BL_SEQUENCES product table 157
BL_TABLE product table 157
BL_TABLEPART product table 157
BL_TABLESPACE product table 157
BL_TRIG_COL product table 157
BL_TRIG_TEXT product table 157
BL_TRIG_TEXT_AUX product table 157
BL_TRIGGER product table 157
BL_UNQCNST product table 157
BL_UNQCNST_COL product table 157
BL_VIEW product table 157
BL_VIEW_TEXT product table 157
BL_VIEW_TEXT_AUX product table 157
BL_VOLS product table 157
BLRP data set 193
BMC Software, contacting 2
BMCDB2 CLIST 138

C
CANCEL ISPF command 55
carriage return, hexadecimal format 145
CASE tools 28
catalog baseline profiles, user option for CM/PILOT 178
CD_ALIAS product table 157
CD_AUXTABLE product table 157
CD_CHECK product table 157
CD_CHECK2 product table 157
CD_COLUMN product table 157
CD_DATABASE product table 158
CD_INDEX product table 158
CD_KEY product table 158
CD_KEYTARGETS product table 158
CD_REL product table 158
CD_RELKEY product table 158
CD_ROUTINES product table 158
CD_SEQUENCES product table 158
CD_STOGROUP product table 158
CD_SYNONYM product table 158
CD_TABLE product table 158
CD_TABLEPART product table 158
CD_TABLESPACE product table 158
CD_TRIG_COL product table 158
CD_TRIG_TEXT product table 158
CD_TRIG_TEXT_AUX product table 158
CD_TRIGGER product table 158
CD_UNQCNST product table 158
CD_UNQCNST_COL product table 158
CD_VIEW product table 158
CD_VIEW_TEXT product table 158
CD_VIEW_TEXT_AUX product table 158
CD_VOLS product table 158
CDL (Change Definition Language)
data set name, specifying 173
data set, allocating 141
CHANGE MANAGER
architecture 37
component of Database Administration 50
components 40
database access 160
Execution plans 147
objects 43
overview 25
plans, list of 146
CHANGE MANAGER objects
application 49
baseline 44
CDL file 48
CM/PILOT worklist 50
DDL file 48
DML 49
internal table 48
profile 44
script 49
task ID 49
unload data set 37
work ID 35
worklist 36
change migration process 26
change rules, used in outbound migrate profiles 47
changes
managing with ALTER 22
managing with CHANGE MANAGER 25
characters
delimited identifiers 144
nonprintable 145
nonviewable 145
check constraints, INSERT statement 161
cleanup, data set 186
CM/PILOT component
    of CHANGE MANAGER 42
CM/PILOT user options
    catalog baseline profile 178
    CDL 179
    changing 176
    data sets 177
    DDL baseline profile 178
    processing a worklist 176
    worklist 179
CM/PILOT worklist 50
CM_APPLICATION product table 158
CM_BASELINE product table 158, 159
CM_BLPROFILE product table 158
CM_LOCATION product table 158
CM_MIGPROFILE product table 158
CM_RENAME product table 158
CM_RULES product table 158
CM_SCOPE product table 158
CM_SECURITY product table 158
CM_SYNC product table 158, 159
CM_TASKID product table 159
CM_UNLOADDSN product table 158
CM_WORKID product table 158
columns, INSERT statement 161
commands
    DOPTS 139
    ENVI 54
    ENVIRONMENT 54
    MAINT 54
Compare component
    default data set name 173
    of CHANGE MANAGER 41
comparisons
    primary input 41
    secondary input 41
    setting default processing options 172
components
    Analysis 33
    Baseline 40
    CM/PILOT 42
    Compare 41
    Execution 34
    Front End 32
    Import 33, 41
    JCL Generation 35
    Specification 32
control characters, using in object names 145
control point 41
conventions, documentation 16
CP_DML_STMT product table 158
CP_SCRIPT product table 158
CP_STEP product table 158
creating
    additional installation options module 139
    objects 57
Cross-System Image Manager. See XIM
customer support 3

D
data conversions, using in change process 24
data modeling tools, using 28
data sets
    $AJXDOC 154
    CDL 141
    cleanup 186
    DATAWORK 191
    deleting automatically 186
diagnostic output 141
discard 187
error 187
JCL 141
LOGSORT 191
map 187
permanent work 193
prefixes 195
punch 187
report 141
resolving names 197
sample VSAM object 185
setting default processing options 193
setting installation options 166
sizing 155, 185
SORTOnnn 187
SORTOUT 187, 193
SORTPnnn 187
SORTWORK 191
SUTnnn 187
SYSCOPY 193
SYSDISC 193
SYSERR 193
SYSMAP 193
SYSPUNCH 193
SYSREC 193
SYSUT 193
SYSUTnnn 187
unload 187
user options, task ID 177
work 187
worklist 141
WRKnnn 187
data structures, importing 25
Database Administration 50
databases, INSERT statement 161
DATASETSIZING B keyword 185
DATASETSIZING C keyword 185
DATASETSIZING O keyword 185
DATAWORK data set 191
DB2 objects, security 160
DB2 SSID, specifying 167
DB2 utilities
used by ALTER 23
used by CHANGE MANAGER 23

DDL
baseline 44
baseline profile 45
debugging, setting default processing options 203
default options. See installation options
defaults, setting user options
  Analysis processing 167
  Baseline processing 174
  Baseline Report processing 175
  Compare processing 172
data set allocation 166
data sets 193
debugging 203
  Execution processing 171
general 135
generation data groups 202
Import processing 169
JCL jobcard 180
JCL static data sets 184
JCL STEPLIBs 182
online reorganizations 208
product options file 214
screen display 164
SHRLEVEL CHANGE 208
SORTWORK data sets 191
tapes 188
user variables 213
utility module names 207
worklist parallelism 209
worklist sorting 211
deleting
  baselines 159
data sets automatically 186
task IDs 159
work IDs 159
delimited identifiers
  valid and invalid wildcard use 143
  wildcard support 144
diagnostic output
data sets 141
default data set name, for Analysis 168
default data set name, for Baseline 174
default data set name, for Baseline Report 176
default data set name, for Compare 173
default data set name, for Execution 171
default data set name, for Import 170
discard data set 187
documentation information 16
DOPTS command 139
DOWN ISPF command 55
DSECTs, AEXUSERX 149
dynamic allocation
data set prefixes 195
diagnostic output data set 141

e
edit macros, AJXPODAT 216
electronic documentation 15
END ISPF command 55
enquiry character, hexadecimal format 145
ENTER ISPF command 55
ENVI command 54
ENVIRONMENT command 54
error data set 187
Execution component
default data set name 171
  of ALTER and CHANGE MANAGER 34
  packages 148
  plans, list of 147
  security exit, function codes 149
  security exit, testing 152
  security exit, using 149
  execution, worklist
    authorization switching 149
    cleanup job step for data sets 186
    default run type 173
    deleting data sets automatically 186
    overriding defaults for worklist parallelism 209
    setting default processing options 171
    external hexadecimal format 145

f
F1 (HELP) 55
F10 (LEFT) 56
F11 (RIGHT) 56
F12 (PREV) 56
F2 (SPLIT) 56
F3 (END) 55
F4 (ZOOM) 56
F7 (UP) 56
F8 (DOWN) 55
F9 (SWAP) 56
Fast Path Navigation commands 54
FCPY_SUPPRESS_SUFF POF keyword 196
fix, using the MAINT command 54
foreign keys, INSERT statement 162
form feed character, hexadecimal format 145
Front End component
data set allocation 141
description 32
generating input streams 139

g
GDG (generation data group)
  setting default processing options 202
  specifying 196
GDG (generation data group) (continued)
   symbolic variable 196
   using for data sets 196
general resource profile 153
generation data group. See GDG
GLID (global authorization ID), default 169

H
HELP ISPF command 55
Help, online 15

I
IEFUSI exit 154
Import component
   default data set name 170
   of ALTER 41
   of CHANGE MANAGER 33
importing
   default run type 170
   setting default processing options 170
indexes, INSERT statement 161
initial POF, refreshing 218
input streams 136
installation options
   creating 139
   POFDS 137
   user options 139
internal format 145
ISPF file tailoring 154
ISPF interface
   AJXPODAT macro 216
   commands 55
   help 55
   Main Menu 53
   using 52
ISPF profile
   dataset 136
   modifying values 136
   user options 139
ISPF skeletons, modifying 154

J
JCL
   data set, allocating 141
   job statement, defaults 180
JCL data set name
   Analysis 168
   Baseline Report 176
   Compare 173
   Execution 171
   Import 170
JCL generation
   data set sizing 155
   product options 137
   user options 137
   VSAM object sampling 155
JCL Generation component 35
JCL jobcard, setting default processing options 180
JCL static data sets, setting default processing options 184
JCL STEPLIBs, setting default processing options 182
JCLCLEANUP keyword 186
-JCLP worklist command 198

L
LEFT ISPF command 56
literal strings 145
LOB SYSREC data set 193
locations, using 48
LOGSORT data set 191

M
macros, AJXPODAT 216
MAINT command 54
maintaining, sync tables 159
maintenance, MAINT command 54
managing change
   using ALTER 22
   using CHANGE MANAGER 25
map data set 187
MAXINITS AEXPIN keyword 210
MEMLIMIT system parameter 154
message files
   AJXPOFER 219
   AJXPOFVL 219
message identifier 57
migrate-type work ID 35
migration
   inbound migrate profile 178
   objects 80
   profile 45
MININITS AEXPIN keyword 210
modifying, user options 179
msg_ID 57
MSSID symbolic variable 198
multitasking
   prefixes for dynamically-allocated data sets 195
   using tape stacking 190

N
naming objects 141
negative acknowledgegment, hexadecimal format 145
NEXT ISPF command 56
nonprintable characters 145
nonviewable characters 145
null character, hexadecimal format 145
null characters 145

O

objects
ALTER 35
CHANGE MANAGER 43
creating 57
migrating 80
naming 141
security 141
VSAM 155

online Help 15
online reorganizations 208

options
Analysis 167
Baseline 174
Baseline Report 175
changing 135
CM/PILOT 176
Compare 172
data set allocation 166
data sets 193
debugging and display 203
Execution 171
generation data group 202
Import 169
installation 135
JCL jobcards 180
JCL static data sets 184
JCL STEPLIBs 182
online reorg 208
panel display 164
product options file 214
refreshing 137
setting 163
SHRLEVEL CHANGE 208
SORTWORK data sets 191
tapes 188
used by products 140
user 136
user variables 213
utility module names 207
worklist parallelism 209
worklist sorting 211
outbound migrate profiles, used by CHANGE MANAGER 46
override options, using keywords 139

P

packages, Execution component 148
panel display 164

PARALLEL AEXPIN keyword 210
parameters
MEMLIMIT 154
system 154
PCPY1_SUPPRESS_SUFF POF keyword 196
PCPY2_SUPPRESS_SUFF POF keyword 196
percent character, hexadecimal format 145
percent sign (%) as a wildcard 142
permanent work data sets 193
PFSHOW ISPF command 56
plans
ALTER 146
authorizations 145
CHANGE MANAGER 146, 147
Execution component 147
names 145
security 145
synonyms 54

POF (product options file)
AJXPODAT macro 216
creating a user POF 214
creating OUTPUT or TEMPLATE descriptors 223
generating reports 219
initial 137
initializing 137
overriding values in SLIBs 221
POF Validation Report 219
populating 137
refreshing the initial POF 218
reusing in a subsequent installation 219
setting default processing options 214
updating a user POF 216
user 138
using multiple POFS 217
Variables Initialized with Default report 219

POFDATE AJXPOFIN keyword 138
POFDS installation option 137
POFDS keyword 137
prefixes 195
PREV ISPF command 56
product database 160
product options file.  See POF
product support 3
product tables
list of 157
maintaining 156
security 160
sizing 159
profiles
baseline 45
catalog baseline 45
DDL baseline 45
inbound migrate 45
migrate 45
outbound migrate 46
user options, application 178
worklist baseline 45
propagating, changes 20
punch data set 187

Q
question mark (?) as a wildcard 142

R
RACF (IBM Resource Access Control Facility) 153
RCPY1_SUPPRESS_SUFF POF keyword 196
RCPY2_SUPPRESS_SUFF POF keyword 196
receive-type work ID 36
receiving SSID, specifying 199
refreshing user options 137
rename table
  security 160
  updating 160
report data set 141
reports
  generating POF 219
  setting installation options 176
restricting access
  specifying a resource profile 153
  to worklist parallelism 152
RIGHT ISPF command 56
ROWID SYSREC data set 193, 200

S
scope rules, CHANGE MANAGER object 46
scope, used by Compare 47
script, CM/PILOT 42
security
  authorization switching 149
  BMC object 141
  component 145
  DB2 objects 160
  Execution exit 149
  Execution security exit function codes 149
  plan 145
  rename table 160
  testing the execution 152
  used by execution 149
sequences, INSERT statement 162
SHRLEVEL CHANGE
  overriding defaults 208
  setting default processing options 208
Simple Space Estimation (SSE) feature 56
skeleton library, overriding POF values 221
SMFPRM member 154
SMS (Storage Management Subsystem) 144, 201
SORTOnnn data set 187
SORTOUT data set 187, 193
SORTPnnn data set 187
SORTWORK data set 191
Space Estimation
  ALTER 25
  CHANGE MANAGER 32
Specification, component of ALTER and CHANGE MANAGER 32
SPLIT ISPF command 56
SSE ISPF command 56
STEPLIB libraries, setting default values 181
storage group, INSERT statement 161
Storage Management Subsystem. See SMS
storage, virtual 154
support, customer 3
SUTnnn data set 187
SWAP ISPF command 56
switching authorization 149
sync table 159
synonyms
  ENVI command 54
  INSERT statement 162
syntax statement conventions 18
SYSCOPY data set, setting JCL options 193
SYSDISC data set 193
SYSERR data set 193
SYSMAP data set 193
SYSPUNCH data set 193
SYSREC data sets
  LOB 193
  ROWID 193
  setting JCL options 193
SYSUT data set 193
SYSUTnnn data set 187

T
table spaces, INSERT statement 161
tables
  baseline 159
  column, INSERT statement 161
  INSERT statement 161
  maintenance 156
  security 160
  sizing 159
tapes
  setting default processing options 188
  stacking, disabling 190
  using 199
target SSID, specifying 199
task IDs
  CHANGE MANAGER object 49
  deleting 159
  user options 177
Task List 58, 81
technical support 3
temporary work data sets 191
threshold value, for DASD 200
threshold, for alternate unit 201
TRACE AEXPIN keyword 210
triggers, INSERT statement 162
TSO exit, to set default option 181
TSSID symbolic variable 198

underscore (_)
   as a literal 143
   as a wildcard 142
unique constraints, INSERT statement 161
unit name, ROWID SYSREC data set 200
unload data set, used by utilities 187
UP ISPF command 56
user exits, Execution security 149
user options
   CDL 179
   changing CM/PILOT 176
   general 135
   JCL Generation 137
   override options 139
   saving in profile 136
user POF
   creating 214
   updating directly 216
   updating in options panels 216
   using multiple 217
user variables
   character 213
   numeric 213
   setting default processing options 213
utility module names, setting default processing options 207
utility template switching 201

values
   hexadecimal 145
   nonprintable characters, using 145
   nonviewable characters, using 145
variables, modifying 154
view column, INSERT statement 162
viewing, INSERT statement 162
virtual storage 154
VSAM object
   data set sizing 185
   JCL generation 155
   sampling used by JCL Generation 155

wildcard characters
   % (percent sign) 142
   * (asterisk) 142
   ? (question mark) 142
   _ (underscore) 142
delimited identifier support 144
   used with change rules 143
   used with scope rules 142
   used with Specification filters 143
valid and invalid wildcards 143
work data sets
   permanent 193
   temporary 191
   used by utilities 187
work IDs
   alter-type 35
   deleting 159
   migrate-type 35
   name template 179
   receive-type 36
worklist parallelism
   overriding defaults 209
   setting default processing options 209
worklist sorting, setting options 211
worklists
   baseline profile 45
   data set 141
   DDL baseline 44
   overview 36
   setting default data set name 168
   used by Baseline 44
   WRKnnn data set 187
XIM (Cross System Image Manager) 152
XIMGROUP AEXPIN keyword 210
XIMPRC AEXPIN keyword 210
XIMSTART AEXPIN keyword 210
zaps
   applying to restrict access 153
   using the MAINT command 54
   ZOOM command 56, 165
Notes