

Oracle® Enterprise Manager

Getting Started with Oracle Change Management Pack

Release 9.0.1

June 2001

Part No. A88717-01

ORACLE®

Oracle Enterprise Manager Getting Started with Oracle Change Management Pack, Release 9.0.1

Part No. A88717-01

Copyright © 2001, Oracle Corporation. All rights reserved.

Primary Author: Bert Rich

Contributors: Jonathan Bauer, Tina Boisvert, Andy Bodge, Ashish Gor, Luming Han, Steven Huberman, David Kulvete, Dina Rabuck, Yuru Stamas, Venkat Tummalapalli, Shushuai Zhu

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Expert, Oracle Net, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
1 Overview of the Oracle Change Management Pack	
New Features for This Release of Oracle Change Management Pack	1-1
Database Objects Supported by Oracle Change Management Pack Applications	1-3
Understanding Oracle Change Management Pack Objects	1-4
Tracking Applications and Change Applications	1-6
Tracking Applications.....	1-6
Change Applications.....	1-7
Overview of Change Manager	1-8
Overview of Create Baseline	1-8
Capturing a Baseline from the Oracle Enterprise Manager Console.....	1-10
Overview of Compare Database Objects	1-10
Performing a Comparison from the Oracle Enterprise Manager Console.....	1-11
Overview of Synchronization Wizard	1-12
Overview of DB Quick Change	1-13
Overview of DB Alter	1-14
Overview of DB Propagate	1-14
Copying Definitions Using Drag and Drop.....	1-15
Overview of Plan Editor	1-16
Comparison of Other Change Applications and Plan Editor	1-17
Starting the Oracle Change Management Pack Quick Tour	1-18
Sample Uses for the Oracle Change Management Pack	1-18

Using Change Plans to Make Changes	1-21
Planning and Defining Changes.....	1-21
Evaluating the Impact of Changes	1-22
Implementing the Changes	1-23
Learning More About Making Changes Using Change Plans.....	1-24
Using Help	1-25
Displaying the Contents Page for Help.....	1-25
Displaying a Help Topic for Your Current Context in an Application	1-25
Finding a Particular Type of Help Topic.....	1-26

2 Using Change Manager

Running Change Manager Connected to an Oracle Management Server or Standalone	2-1
Creating a Standalone Repository for Change Management Pack Applications	2-2
Changing Your Standalone Change Management Pack Repository.....	2-2
Starting Change Manager	2-3
Using the Change Manager Main Window	2-4
Starting Change Management Pack Applications from Change Manager	2-6
Working with Baselines	2-7
Working with Comparisons	2-8
Working with Change Plans	2-10
Viewing History Entries for Tasks	2-11
Viewing and Completing Pending Tasks.....	2-12
Viewing Completed Tasks	2-13
Working With a Particular Version of a Change Plan, Baseline, or Comparison	2-13
Exporting and Importing Change Plans, Baselines, and Comparisons	2-14

3 Using Plan Editor

Modifying and Creating Object Definitions with Plan Editor	3-2
Creating a Plan with Plan Editor	3-3
Understanding Change Requests.....	3-4
Understanding Directives	3-4
Understanding Scoped Directives.....	3-5
Understanding Exemplars	3-5
Understanding Modified Exemplars.....	3-5
Defining Change Requests with Plan Editor	3-6

Defining a Directive with Plan Editor	3-7
Defining a Scoped Directive with Plan Editor	3-8
Defining an Exemplar with Plan Editor	3-10
Creating a Modified Exemplar with Plan Editor	3-13
Selecting a Destination Database with Plan Editor	3-14
Understanding Script Generation	3-14
Generating a Script with Plan Editor	3-15
Viewing the Impact Report and Script Summary with Plan Editor	3-18
Understanding Script Execution	3-19
Executing the Script with Plan Editor	3-20
Dealing with Script Execution Errors	3-22

A Command Line Interface Appendix

Possible Uses for the Command Line Interface	A-1
Understanding the Notation Used for the Command Syntax	A-2
Requirements for Using the Command Line Interface	A-2
Understanding Case-Sensitivity Requirements for the Command Line Interface	A-3
Using the Command Line Interface with the Oracle Management Server	A-3
Scheduling the Execution of Command Line Interface Scripts	A-4
Using the Command Line Interface Commands	A-4
login command	A-5
logout command	A-7
capture command	A-7
compare command	A-8
generate command	A-9
execute command	A-10
Specifying the Direction in Which to Execute the Script	A-11
Specifying Completion Actions for a Script Execution	A-12
Examples of Using the Execute Command	A-13
Status Codes Returned by the Command Line Interface	A-14

B Troubleshooting Appendix

Using Environment Variables When Running Change Manager from the Command Line	B-1
--	------------

Enabling Tracing and Debugging for Oracle Change Management Pack..... B-2

Index

Send Us Your Comments

Oracle Enterprise Manager Getting Started with Oracle Change Management Pack, Release 9.0.1

Part No. A88717-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc-doc_us@oracle.com
- FAX: 603-897-3317 Attn: Oracle Change Management Pack Documentation
- Postal service:
Oracle Corporation
Oracle Change Management Pack Documentation
One Oracle Drive
Nashua, NH 03062
U.S.A.

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please visit the support website at Oracle MetaLink (<http://www.oracle.com/support/metalink/index.html>).

Preface

This section describes the purpose and organization of this guide: *Getting Started with Oracle Change Management Pack*. Specifically, it covers the following topics:

- [Purpose of this Guide](#)
- [Audience](#)
- [How this Guide is Organized](#)
- [How to Use this Guide](#)
- [Oracle Enterprise Manager Pack Information](#)
- [Oracle Enterprise Manager Documentation](#)
- [Your Comments are Welcome](#)
- [Oracle Support](#)
- [Documentation Accessibility](#)

Purpose of this Guide

This guide provides an overview of the Oracle Change Management Pack applications and their features. As the guide describes how to use the Oracle Change Management Pack applications, it also introduces you to key Oracle Change Management Pack concepts and terminology.

Audience

This guide is written for those who want to use any of the Oracle Change Management Pack applications.

Knowledge Assumed of the Reader

This guide assumes you are familiar with the Oracle Enterprise Manager Console and base applications. If you are not, please read the *Oracle Enterprise Manager Concepts Guide* and the *Oracle Enterprise Manager Administrator's Guide*.

How this Guide is Organized

This guide consists of three chapters and two appendices.

[Chapter 1, "Overview of the Oracle Change Management Pack"](#)

This chapter provides an overview of Oracle Change Management Pack applications, concepts, and features.

[Chapter 2, "Using Change Manager"](#)

This chapter describes how to use Change Manager, the central application for the Oracle Change Management Pack.

[Chapter 3, "Using Plan Editor"](#)

This chapter describes how to modify and create object definitions using the Plan Editor application of Oracle Change Management Pack.

[Appendix A, "Command Line Interface Appendix"](#)

This appendix describes uses for the command line interface for the Oracle Change Management Pack. It also provides syntax and reference information for command line interface commands and describes the exit status codes returned by the command line interface.

[Appendix B, "Troubleshooting Appendix"](#)

This appendix describes different ways of troubleshooting problems that may occur when you are using Oracle Change Management Pack applications.

How to Use this Guide

This guide should be read in its entirety to provide you with an overview of each of the Oracle Change Management Pack applications, so that you are familiar with the tasks that can be performed with them. This guide does not describe all the details of using the applications. The online help for the applications provides more detailed information about using these applications.

Oracle Enterprise Manager Pack Information

The Oracle Enterprise Manager product family includes the following packs: Oracle Change Management Pack, Oracle DBA Management Pack, Oracle Diagnostics Pack, Oracle Management Pack for Oracle Applications, Oracle Management Pack for SAP R/3, Oracle Standard Management Pack, and Oracle Tuning Pack. Each pack is fully integrated into the Oracle Enterprise Manager Console framework.

Oracle Change Management Pack

- Includes Change Manager, Compare Database Objects, Create Baseline, DB Alter, DB Propagate, DB Quick Change, Synchronization Wizard, and Plan Editor.
- Tracks metadata changes in databases.
- Eliminates errors and loss of data when upgrading databases to support new applications.
- Analyzes the impact and complex dependencies associated with metadata change and automatically performs upgrades.
- Allows you to find the database objects that match a set of search criteria that you specify.
- Initiates change safely with easy-to-use wizards that teach the systematic steps necessary to upgrade.

Oracle DBA Management Pack

- Simplifies DBA operations by allowing you to perform them by pointing and clicking with your mouse rather than manually entering SQL commands.
- Concurrently performs multiple tasks in multiple Oracle Enterprise Manager windows.
- Administers several databases simultaneously.
- Centralizes database administration tasks for both local and remote databases running on any Oracle platform in any location worldwide.

Oracle Diagnostics Pack

- Includes Oracle Performance Manager, Oracle Capacity Planner, Oracle Trace, Oracle TopSessions, and Oracle Event Tests that are provided with the Oracle Diagnostics Pack.

- Monitors, diagnoses, and maintains the health of databases, operating systems, and applications. Real-time analysis is used to analyze current problems, and historical analysis is used to analyze trends that could cause future problems.
- Provides powerful capacity planning features that enable users to easily plan and track future system resource requirements.

Oracle Management Pack for Oracle Applications

- Includes Oracle Performance Manager, Oracle Capacity Planner, and Oracle Event Tests that are provided with the Oracle Management Pack for Oracle Applications.
- Monitors, diagnoses, and maintains the health of Oracle Applications. Real-time analysis is used to analyze current problems, and historical analysis is used to analyze trends that could cause future problems.
- Provides powerful capacity planning features that enable users to easily plan and track future system resource requirements.

Oracle Management Pack for SAP R/3

- Includes Oracle Performance Manager, Oracle Capacity Planner, and Oracle Event Tests that are provided with the Oracle Management Pack for SAP R/3.
- Monitors, diagnoses, and maintains the health of an SAP R/3 system.
- Provides powerful capacity planning features that enable users to easily plan and track future system resource requirements. Real-time analysis is used to analyze current problems, and historical analysis is used to analyze trends that could cause future problems.

Oracle Standard Management Pack

- Includes Create Baseline, Compare Database Objects, Change Manager, Oracle Index Tuning Wizard, Oracle Performance Manager, and Oracle TopSessions.
- Monitors and diagnoses problems, tunes high impact indexes, and tracks and compares changes in your Oracle environment.

Oracle Tuning Pack

- Includes Oracle SQL Analyze, Oracle Expert, Oracle Index Tuning Wizard, Reorg Wizard, and the Tablespace Map.
- Optimizes system performance by identifying and tuning major database and application bottlenecks such as inefficient SQL, poor data structures, and improper use of system resources.

- Proactively discovers tuning opportunities and automatically generates the analysis and required changes to tune the system. Inherent in the product are powerful teaching tools which train DBAs how to tune as they work.
- Instills "Consultant Quality" tuning expertise into developers and DBAs and increases their productivity.

Oracle Enterprise Manager Documentation

The *Getting Started with Oracle Change Management Pack* manual (this manual) is one of several Oracle Enterprise Manager documents.

Oracle Enterprise Manager base documentation

- The *Oracle Enterprise Manager Readme* provides important notes regarding the online documentation, updates to the software, and other late-breaking information for Oracle Enterprise Manager and the Oracle Management Packs.
- The *Oracle Enterprise Manager Concepts Guide* provides an overview of Oracle Enterprise Manager, including the DBA Management Pack.
- The *Oracle Enterprise Manager Administrator's Guide* explains how to use Oracle Enterprise Manager, Oracle's systems management console, common services, and integrated platform tool. This manual also describes how to use the DBA Management Pack.
- The *Oracle Enterprise Manager Configuration Guide* explains how to configure Oracle Enterprise Manager.
- The *Oracle Enterprise Manager Messages Manual* describes the Oracle Enterprise Manager error messages and methods for diagnosing the messages.

Oracle Enterprise Manager and Management Pack Installation documentation

- The *Oracle 9i Installation Guide Release 9.0.1* provides important information for installing the Oracle Enterprise Manager Console and the Oracle Management Packs. This manual also contains information such as hardware and software requirements, installable components, and deinstallation procedures. The *Oracle 9i Installation Guide Release 9.0.1* is located on the Oracle9i Online Generic Documentation CD-ROM, Release 1 (9.0.1) or you can get a free download of the manual from the Oracle Documentation Center at <http://docs.oracle.com/>.

Oracle Change Management Pack documentation

- The *Getting Started with Oracle Change Management Pack* manual (this manual) provides an overview of the concepts and features of the Oracle Change Management Pack applications.

Oracle DBA Management Pack documentation

- The *Oracle Enterprise Manager Concepts Guide* provides overview information about the DBA Management Pack and the Oracle Enterprise Manager Console.
- The *Oracle Enterprise Manager Administrator's Guide* provides information on how to use the DBA Management Pack and the Oracle Enterprise Manager Console.

Oracle Diagnostics Pack documentation

- The *Getting Started with the Oracle Diagnostics Pack* manual provides an overview of the concepts and features of the Oracle Diagnostics Pack applications.
- The *Oracle Trace User's Guide* explains how to use the Oracle Trace application to capture and use historical data to monitor Oracle databases.

Oracle Management Pack for Oracle Applications documentation

- The *Getting Started with Oracle Management Pack for Oracle Applications* provides an overview of the concepts and features of the applications in the Oracle Management Pack for Oracle Applications.
- The *Oracle Intelligent Agent User's Guide* provides configuration information and answers to crucial troubleshooting questions pertaining to the Oracle Intelligent Agent.

Oracle Management Pack for SAP R/3 documentation

- The *Oracle Management Pack for SAP R/3 Readme* provides important notes regarding the Management Pack for SAP R/3 online documentation, updates to the software, and other late-breaking information.
- The *Getting Started with Oracle Management Pack for SAP R/3* manual provides an overview of the concepts and features of the Oracle Management Pack for SAP R/3 applications.
- The *Installation Guide for Oracle Enterprise Manager with Oracle Management Pack for SAP R/3* provides important information for installing the Oracle Management Pack for SAP R/3. This manual also contains information such as hardware and software requirements, installable components, and deinstallation procedures.

Oracle Standard Management Pack documentation

- The *Getting Started with Oracle Standard Management Pack* manual provides an overview of the concepts and features of the Oracle Standard Management Pack applications.

Oracle Tuning Pack documentation

- The *Database Tuning with the Oracle Tuning Pack* manual provides an overview of the concepts and features of the Oracle Tuning Pack applications. A description of how these applications can work together to tune an Oracle database is also provided.

Obtaining Oracle Documentation

To download free release notes or installation information, please visit the Oracle Documentation Center at <http://docs.oracle.com/>.

Printed documentation is available for sale in the Oracle Store at <http://oraclestore.oracle.com/>.

Your Comments are Welcome

We value and appreciate your comments as an Oracle user and reader of the manuals. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. After the table of contents in our printed manuals is a Reader's Comment Form, which we encourage you to use to tell us what you like or dislike about this manual or other Oracle manuals. If the form is not available, please use the following address, FAX number, or electronic mail address.

Oracle Corporation
Oracle Change Management Pack Documentation
One Oracle Drive
Nashua, NH 03062
U.S.A.

FAX - 603-897-3317 Attn: Oracle Change Management Pack Documentation
Electronic mail - nedc-doc_us@oracle.com

Oracle Support

The Oracle MetaLink site (<http://www.oracle.com/support/metalink/index.html>) provides access to information that will aid you in using Oracle products such as: hot topics, product reference, installation assistance materials, white papers, problem/solution articles, and more. To use this site you must be a licensed Oracle

user with an active support service contract. Contact your Oracle sales representative for more information.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program web site at <http://www.oracle.com/accessibility/>.

Overview of the Oracle Change Management Pack

The Oracle Change Management Pack is a group of integrated applications used to track and make changes to database object definitions. The Oracle Change Management Pack is a component of Oracle Enterprise Manager.

This chapter provides an overview of Oracle Change Management Pack applications, concepts, and features.

The following section describes new features for this release of Oracle Change Management Pack.

New Features for This Release of Oracle Change Management Pack

The following list describes the major new features for this version of the Oracle Change Management Pack:

- Exemplars can be modified

In previous Oracle Change Management Pack releases, you could not modify the attributes of the exemplar definition before propagating the definition to a new destination. Instead, you had to modify the definition in the new location after the propagation was complete.

In this release, you can edit the attributes of an exemplar definition before propagating it. For example, you can modify the storage parameters for exemplar definitions before propagating the definitions from a production database to a test database. After the propagation operation completes, the propagated definitions will already have the desired storage parameters for the new destination.

An exemplar with attributes that have been changed since the exemplar was created is referred to as a modified exemplar.

See "[Understanding Modified Exemplars](#)" on page 3-5 for more information about modified exemplars.

- Instance parameters can be included in baselines

The Create Baseline and Compare Database Objects applications support Oracle instance parameters in this release.

You can use Create Baseline to create a baseline that includes definitions and instance parameter settings for a database, or you can create a baseline that contains only the instance parameters for a database.

You can use Compare Database Objects to compare a database's current instance parameter settings to the database's instance parameter settings that were captured in a previous baseline. Or, you can compare two different baselines taken at different points in time that include a database's instance parameter settings.

Comparing instance parameters differences for a database can help you monitor changes in the database's configuration, which may enable you to diagnose performance problems.

- Change Management Pack applications can be run standalone (without a connection to the Oracle Management Server)

In previous releases, Change Management Pack applications could only be run after connecting to an Oracle Management Server. With this release, you can run Change Management Pack applications standalone (without a connection to the Oracle Management Server). See "[Running Change Manager Connected to an Oracle Management Server or Standalone](#)" on page 2-1 for more information.

To run Change Management Pack applications standalone, you must have a standalone repository. See the *Oracle Enterprise Manager Configuration Guide* for information on the steps to perform before creating a standalone repository.

- Change Management Pack applications run in a web browser

In this release, you can run Change Management Pack applications from a web browser. To do this, you start the Oracle Enterprise Manager Console in a browser, then choose **Tools->Change Management Pack->Change Manager** to start Change Manager in the browser. Other Change Management Pack applications that you start from Change Manager also run in the browser.

The majority of the Change Management Pack features are available from a browser. However, you cannot perform import and export operations from the Change Manager application when it is running in a browser. Also, you cannot use the command line interface commands from a browser.

For information on running the Oracle Enterprise Manager Console in a web browser, see the *Oracle Enterprise Manager Configuration Guide*.

- Change Management Pack scripts can be submitted more easily to the Oracle Enterprise Manager job system

To submit a change plan script as a job in previous versions of Change Management Pack, you needed to exit the Change Management Pack application and use the Oracle Enterprise Manager Console's job system. In this release, Change Management Pack applications provide a job scheduling interface that lets you to schedule the submission of a script to the job system without exiting the Change Management Pack application.

- The command line interface is available on UNIX

In previous releases, the Oracle Change Management Pack command line interface was available for the Windows NT operating system. In this release, the command line interface is also available for UNIX operating systems.

See [Appendix A, "Command Line Interface Appendix"](#) on page A-1 for more information about the command line interface.

Database Objects Supported by Oracle Change Management Pack Applications

The types of database object definitions that Oracle Change Management Pack supports are:

- Cluster
- Database link
- Function
- Index
- Materialized view
- Materialized view log
- Package

- Package body
- Procedure
- Profile
- Refresh group
- Role
- Rollback segment
- Sequence
- Synonym
- Table
- Tablespace
- Trigger and instead-of trigger
- User
- View

Oracle Change Management Pack currently does not support the following features of the database objects in the previous list:

- Object-oriented features:
 - Tables and snapshot logs that refer to user-defined data types.
 - Object tables
 - Nested tables
 - Object views
- Snapshots that are not read-only

For a more complete list that describes the level of support Oracle Change Management Pack offers for specific database objects and attributes, see the section on currently unsupported database features and attributes in the Change Management Pack section of the *Oracle Enterprise Manager Readme*.

Understanding Oracle Change Management Pack Objects

The following types of objects are created only by Oracle Change Management Pack applications and are used to make changes or track changes to other object definitions:

- **Change plans**

You can use several Oracle Change Management Pack applications, including DB Quick Change, DB Alter, DB Propagate, Plan Editor, and the Synchronization Wizard, to create change plans. A change plan is an object that serves as a container for change requests. From a change plan, you can generate one or more scripts that can be executed to carry out the changes specified in the change plan at one or more destination databases. Oracle Change Management Pack applications do all the work of determining and ordering the individual steps that must be carried out to execute the change requests in the plan. After a script executes at a destination database, you have the option of keeping or undoing the changes made by the script. See "[Using Change Plans to Make Changes](#)" on page 1-21 for more information about change plans.

- **Baselines**

A baseline is a group of database object definitions captured by the Create Baseline application at a particular point in time. Each baseline has the name of its baseline specification and a version number. A baseline is stored in a format that can be used by multiple Oracle Change Management Pack applications.

A baseline specification is used to generate baselines. A baseline specification includes a baseline name, source database, scope specification, and owner. The baseline name can be changed after the baseline specification is created, but the source database, scope specification, and owner cannot. The scope specification for a baseline is a set of criteria that an object must meet to be included in the baseline generated from the baseline specification. The scope specification describes the types of database object definitions to be included in the baseline, the schemas in which to look for those object definitions, and the names of those object definitions. See "[Overview of Create Baseline](#)" on page 1-8 for more information about baselines.

- **Comparisons**

A comparison identifies the differences found by the Oracle Change Management Pack in two sets of database object definitions that you have specified in the Compare Database Objects application. Each comparison has the name of its comparison specification and a version number. Comparison results are displayed in the Comparison Viewer.

A comparison specification is used to generate comparisons. A comparison specification is defined by its name, source database, scope specification, and owner. The name can be changed after the comparison specification is created, but the source database, scope specification, and owner cannot. The scope specification for a comparison is a set of criteria that the set of objects to be

compared must meet to be included in a comparison generated from the comparison specification. The scope specification describes the types of database object definitions to be included in the comparison, the schemas in which to look for those object definitions, and the names of those object definitions. See "[Overview of Compare Database Objects](#)" on page 1-10 for more information about comparisons.

Tracking Applications and Change Applications

Oracle Change Management Pack includes applications that let you track changes and applications that let you make changes.

The Change Manager application is the central interface for the Oracle Change Management Pack. Change Manager provides direct or indirect access to all Oracle Change Management Pack features. For example, Change Manager provides direct access to the Oracle Change Management Pack history features. It provides indirect access to the other Oracle Change Management Pack applications, which can be started from Change Manager. Because it provides access to all Change Management Pack features, Change Manager can be thought of as both a tracking and change application. See "[Starting Change Manager](#)" on page 2-3 for more information on starting Change Manager and "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for more information on starting other applications from Change Manager.

The next two sections explain how you can track changes with the tracking applications and how you can make changes using the change applications.

Tracking Applications

With Oracle Change Management Pack, you can track changes by:

- Capturing the current definitions of a set of database objects in a baseline.
- Creating new versions of a baseline at regular intervals to track changes in that set of database objects over time.
- Comparing a set of object definitions in one schema, database, or baseline to another set of definitions (in a schema, database, or baseline). A comparison identifies differences between two sets of definitions.
- Creating new versions of a comparison at regular intervals to determine the differences between the two sets of definitions over time.

The tracking applications and their functions are:

- **Create Baseline**
Captures one or more definitions from one database.
- **Compare Database Objects**
Compares a set of object definitions in one schema, database, or baseline to another set of definitions (in a schema, database, or baseline).

Change Applications

With Oracle Change Management Pack applications, you can make changes by:

- **Synchronizing one set of definitions to match the other set (after creating a comparison).**
- **Reproducing one or more database object definitions at multiple databases, which has the same effect as copying definitions from one database to another. If an object definition of the same name and type already exists at the destination database, that definition is modified to match the source definition.**
- **Modifying one or more object definitions in a database by specifying the changes to make.**

The change applications and their functions are:

- **Synchronization Wizard**
Modifies a set of database definitions to match another set of database definitions.
- **DB Quick Change**
Modifies one definition in one database.
- **DB Alter**
Modifies one or more definitions in one or more databases.
- **DB Propagate**
Reproduces one or more definitions from a database in another schema in that database or in another database.
- **Plan Editor**
Allows you to create a change plan, which serves as a container for change requests. Scripts can be generated from a change plan, then executed on one or more databases to make the changes specified by the change requests.

The following sections provide additional overview information on the Oracle Change Management Pack applications.

Overview of Change Manager

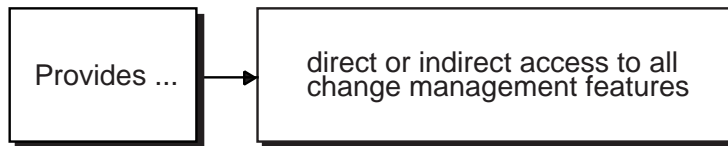
The Change Manager application is the central interface for the Oracle Change Management Pack. Change Manager gives you direct or indirect access to Oracle Change Management Pack operations.

Using Change Manager, you can view Oracle Change Management Pack objects (change plans, baselines, and comparisons), as well as check on the status of current tasks. You can also view history information that tells you which Oracle Change Management Pack operations were performed by what users at what time.

You can also use Change Manager to start the other Oracle Change Management Pack applications.

[Figure 1-1](#) shows the functional capabilities of Change Manager.

Figure 1-1 Functional Capabilities of Change Manager



NU-3705A-AI

See [Chapter 2, "Using Change Manager"](#) on page 2-1 for more information on using Change Manager. The chapter includes information on starting Change Manager, starting other applications from Change Manager, and on the features that are accessible from the Change Manager application.

Overview of Create Baseline

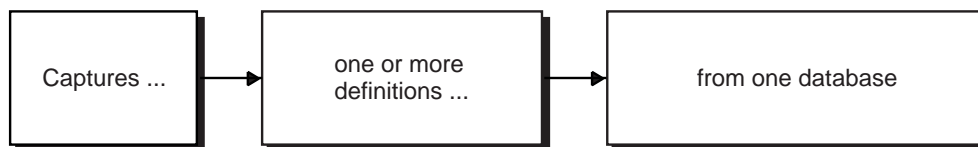
The Create Baseline application guides you step-by-step through the process of capturing a database (or a subset of a database) in a form that is readable by other Oracle Change Management Pack applications. Create Baseline lets you specify the set of database object definitions to capture, captures those definitions in their current state at the time of the capture operation, and displays the baseline in the Baseline Viewer.

Create Baseline can capture object definitions in both of the following forms:

- A baseline. The baseline stores definitions in a form that other Oracle Change Management Pack applications can use. A baseline is created during every capture operation.
- A SQL DDL script. The script contains the SQL statements that correspond to the captured definitions. You can use the script to save a textual version of the definitions or as input to CASE tools that accept SQL DDL input. You can also use individual SQL statements from the script to create definitions in a new database. You have the option of whether or not to create a SQL DDL script. To save time during a capture operation, you can defer the creation of the SQL DDL script.

Figure 1–2 shows the functional capabilities of Create Baseline.

Figure 1–2 Functional Capabilities of Create Baseline



NU-3699A-AI

You can start Create Baseline from Change Manager. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting Create Baseline from Change Manager.

The Baseline Viewer displays baselines created with Create Baseline. You can also use Baseline Viewer to generate a SQL DDL script for the baseline if you did not generate one when the baseline was captured. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting Baseline Viewer from Change Manager.

To generate a report that includes information about the objects in a baseline, view the baseline in Baseline Viewer, click the Database Objects folder, and choose **File->List Report**. The online help contains more information about the specifics of using Baseline Viewer.

Capturing a Baseline from the Oracle Enterprise Manager Console

When you have Oracle Change Management Pack installed, you can also perform a capture operation from the Oracle Enterprise Manager Console by following these steps:

1. Select a database, type, or schema folder in the Oracle Enterprise Manager navigator.
2. Click the right mouse button.
3. On the context menu, choose **Change Management->Capture Database Objects**.

Create Baseline is started to capture a baseline for the selected database, then after the capture operation is completed, the Baseline Viewer is started to allow you to view the baseline.

The online help contains more information about the specifics of using Create Baseline.

Overview of Compare Database Objects

The Compare Database Objects application guides you step-by-step through the process of selecting two sets of object definitions for a comparison. The object definitions can be selected from current database definitions or they can be selected from baselines created earlier with Create Baseline. Differences that are found between the two sets of object definitions are displayed in the Comparison Viewer.

The result of comparing two sets of object definitions using Compare Database Objects is called a comparison. You can view a comparison, name it, and save it using the Comparison Viewer.

Figure 1-3 shows the functional capabilities of Compare Database Objects.

Figure 1-3 *Functional Capabilities of Compare Database Objects*



NU-3700A-AI

You can start Compare Database Objects from Change Manager. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting Change Database Objects from Change Manager.

You can also generate a report about the results of a comparison by choosing **File->Report** in Comparison Viewer. The report includes:

- A description of what you compared.
- The comparison options you selected.
- A summary of the comparison results.
- The comparison tree and comparison details (optional).

After two sets of object definitions have been compared, you can use the Synchronization Wizard to modify one set of definitions (with the exception of baselines, which are read-only and cannot be modified) to match the other set of definitions. See "[Overview of Synchronization Wizard](#)" on page 1-12 for more information about the Synchronization Wizard.

Performing a Comparison from the Oracle Enterprise Manager Console

When you have the Oracle Change Management Pack installed, you can also perform a comparison operation from the Oracle Enterprise Manager Console by following these steps:

1. Select a database, type, or schema folder in the Oracle Enterprise Manager navigator.
2. Click the right mouse button.
3. On the context menu, choose **Change Management->Compare Database Objects**.

Compare Database Objects is used to specify the two sets of database object definitions to include in a comparison, and the Comparison Viewer is used to display the results of the comparison. If the comparison results indicate that certain database definitions are different and you would like to synchronize some or all of the definitions, you can start the Synchronization Wizard by choosing **Tools->Synchronization** in Comparison Viewer. Synchronization Wizard allows you to specify how you want to synchronize the definitions.

The online help contains more information about the specifics of using Compare Database Objects, the Synchronization Wizard, and Comparison Viewer.

Overview of Synchronization Wizard

After two sets of object definitions have been compared and the comparison results are displayed in the Comparison Viewer, you can use the Synchronization Wizard to modify one set of definitions (with the exception of baselines, which are read-only and cannot be modified) to match the other set of definitions.

The Synchronization Wizard guides you step-by-step through the process of specifying the definitions to be synchronized. The Synchronization Wizard also does all the work of determining and ordering the individual steps that must be carried out to perform this requested synchronization, taking object definition dependencies into account.

When the definitions are synchronized, the Synchronization Wizard sees whether each selected object definition in the source also exists at the destination. If an object definition of the same name and type as a source definition already exists at the destination database, the destination object is modified to match the source definition. If a selected object definition in the source does not exist at the destination database, it is created.

After you execute a change plan script created by the Synchronization Wizard at a destination database, you have the option of keeping or undoing the changes made by the script.

Figure 1–4 shows the functional capabilities of the Synchronization Wizard.

Figure 1–4 Functional Capabilities of Synchronization Wizard



NU-3701A-AI

To start the Synchronization Wizard, choose **Tools->Synchronization Wizard** from the Comparison Viewer. See "[Working with Comparisons](#)" on page 2-8 for information on viewing a comparison in the Comparison Viewer.

Note: The Compare Statistics option in Compare Database Objects lets you compare index and table statistics, but the Synchronization Wizard does not synchronize these index and table statistics.

Overview of DB Quick Change

The DB Quick Change application guides you step-by-step through the process of making one or more changes to a single database object definition. You indicate changes to object definitions by directly manipulating their representation on property sheets similar to those in the Oracle Enterprise Manager Console. The DB Quick Change versions of these property sheets allow you to make some changes that cannot be made using the Console (for example, changing the name of a column in a table).

DB Quick Change does all the work of determining and ordering the individual steps that must be carried out to perform your requested operations, taking object definition dependencies into account.

After you execute a change plan script created by DB Quick Change, you have the option of keeping or undoing the changes made by the script.

[Figure 1-5](#) shows the functional capabilities of DB Quick Change.

Figure 1-5 *Functional Capabilities of DB Quick Change*



NU-3702A-AI

You can start DB Quick Change from Change Manager. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting DB Quick Change from Change Manager.

The online help contains more information about the specifics of using DB Quick Change.

Overview of DB Alter

The DB Alter application guides you step-by-step through the process of making changes to one or more object definitions in one or more databases. You indicate changes to object definitions by directly manipulating their representation on property sheets similar to those in the Oracle Enterprise Manager Console. The DB Alter versions of these property sheets allow you to make some changes that cannot be made using the Console (for example, changing the name of a column in a table).

When you specify a group of changes to several database object definitions, DB Alter does all the work of determining and ordering the individual steps that must be carried out to perform your requested operations, taking object definition dependencies into account.

The specified changes can be applied at the source database and at other databases.

After you execute a change plan script created by DB Alter at a destination database, you have the option of keeping or undoing the changes made by the script.

Figure 1-6 shows the functional capabilities of DB Alter.

Figure 1-6 Functional Capabilities of DB Alter



NU-3703A-RA

You can start DB Alter from Change Manager. See ["Starting Change Management Pack Applications from Change Manager"](#) on page 2-6 for information on starting DB Alter from Change Manager.

The online help contains more information about the specifics of using DB Alter.

Overview of DB Propagate

The DB Propagate application guides step-by-step you through the process of selecting one or more object definitions from a database, then reproducing those definitions in a destination schema within the source database or in one or more destination databases. For each selected object definition in the source database, if

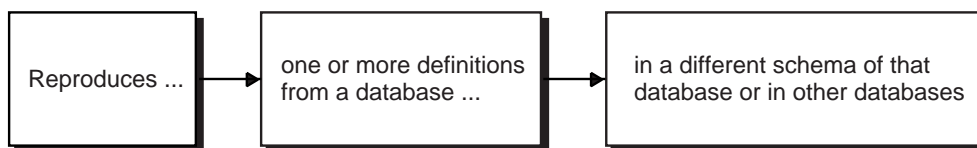
an object definition of the same name and type already exists at the destination database, the destination definition will be modified to match the source definition. If a selected object definition in the source database does not exist at the destination database, it will be created.

DB Propagate does all the work of determining and ordering the individual steps that must be carried out to perform your requested operations, taking object definition dependencies into account.

After you execute a change plan script created by DB Propagate at a destination database, you have the option of keeping or undoing the changes made by the script.

Figure 1-7 shows the functional capabilities of DB Propagate.

Figure 1-7 Functional Capabilities of DB Propagate



NU-3704A-AI

You can start DB Propagate from Change Manager. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting DB Propagate from Change Manager.

The online help contains more information about the specifics of using DB Propagate.

Copying Definitions Using Drag and Drop

When you have the Oracle Change Management Pack installed, you can also copy definitions using the Oracle Enterprise Manager Console. From the navigator pane in the Console, you can select object definitions in a schema, then drag them and drop them into a different schema (which can be in a different database). If you drag a non-schema object definition, it must be dropped in a different database. When you copy table definitions using the drag and drop method, both the table and its associated data is copied to the destination. DB Propagate is launched to complete drag and drop copy operations.

The change plan created by the drag and drop operation is deleted, but a history entry for the copy operation is created and saved. The history entry can be viewed using Change Manager's history features. See ["Viewing History Entries for Tasks"](#) on page 2-11 for more information on using Change Manager to view history entries.

Note: Database links are used to copy table data from one database to another.

If you attempt to copy table data from one database to another and the impact report includes an ERROR-level message about not being able to find a global name for the source database or destination database, this means that a database link must be created for the database or databases referenced in the error message. See ["Viewing the Impact Report and Script Summary with Plan Editor"](#) on page 3-18 for information on impact reports.

See the section on the CREATE DATABASE LINK statement in the *Oracle9i SQL Reference* manual for more information on creating a database link.

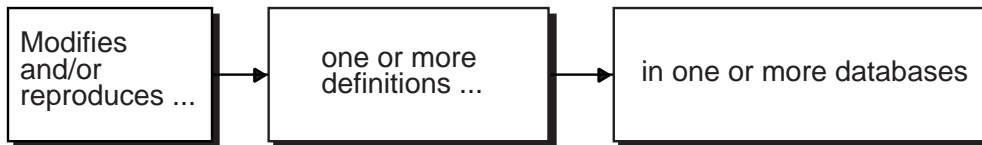
Overview of Plan Editor

The Plan Editor application allows you to create, modify, and deploy a single change plan. The change plan can modify or reproduce database object definitions in one or more databases.

After you execute a change plan script created by Plan Editor at a destination database, you have the option of keeping or undoing the changes made by the script.

[Figure 1-8](#) shows the functional capabilities of Plan Editor.

Figure 1-8 *Functional Capabilities of Plan Editor*



NE-0004A-AI

You can start Plan Editor from Change Manager. See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for information on starting Plan Editor from Change Manager.

Comparison of Other Change Applications and Plan Editor

You can create change plans with Synchronization Wizard, DB Alter, DB Quick Change, and DB Propagate, as well as with Plan Editor. These change applications differ from Plan Editor because each of them is designed to create a change plan with specific types of change requests that make specific types of changes.

Plan Editor is a more flexible change application. It allows you to create and modify a change plan that includes any type of change request and which can make a wider variety of changes. Because Plan Editor plans can include any type of change request, you can use it to edit plans that were created with other Change Management Pack applications and to generate and execute scripts for those plans.

The other applications that create change plans have page-oriented interfaces that guide you step-by-step through the process of specifying the changes you want to make and making those changes. Plan Editor's interface is not page-oriented; it does not guide you step-by-step. You must learn the process of using Plan Editor to create and modify change plans.

Table 1-1 Comparison of Change Applications and Plan Editor

Change Applications	Plan Editor
Can create change plans that include only certain types of change requests.	Can create change plans that include all types of change requests.
Page-oriented interface leads you through the process of creating/modifying a change plan.	Interface is not page-oriented. You must learn the process for creating/modifying a change plan.

Fortunately, learning to use Plan Editor is not difficult. The other change applications use components from the Plan Editor interface. Therefore, as you use the other change applications, you become familiar with the components that Plan Editor uses. Also, [Chapter 3, "Using Plan Editor"](#) on page 3-1 provides complete information on how to use Plan Editor to create change and modify change plans that allow you to create and modify object definitions.

Starting the Oracle Change Management Pack Quick Tour

The Oracle Change Management Pack includes a Quick Tour that provides an overview of the Pack and its components. The Quick Tour is an excellent way to start learning about the applications in the Pack and their uses. To start the Quick Tour, start Change Manager, then choose **Help->Take a Quick Tour**.

Sample Uses for the Oracle Change Management Pack

This section provides some sample uses for Change Management Pack in an application development environment.

Create Baseline Specifications

At the beginning of a new development cycle, use Create Baseline to create one or more baseline specifications that identify the database and schema object definitions used in your application.

Create Baseline Versions at Key Points

You can use a baseline specification to create a baseline that captures the database and schema object definitions used in your application at the start of the development cycle. Then, at regular intervals (such as daily or weekly) and at development milestones (such as beta freeze), create a new version of the baseline that captures the definitions at that point in time. These baseline versions provide a record of the definitions throughout the development cycle.

You can automate the creation of these baseline versions by creating a Change Management Pack command line interface script to capture the baseline and using the facilities of the NT or Solaris operating system to schedule the execution of that script. See [Appendix A, "Command Line Interface Appendix"](#) on page A-1 for more information about the command line interface.

Generate Reports for Definitions in Baselines

Baseline Viewer allows you to view baselines and to generate a report that describes the definitions captured in the baseline.

Generate SQL Statements for Baseline Definitions

You can also use Baseline Viewer to generate the SQL data definition statements for each of the definitions in the baseline, and save these SQL statements to a file. The statements could then be used in a script to reproduce the definitions at some point in the future.

Create Comparison Specifications

Use Compare Database Objects to create comparison specifications that identify two sets of database definitions that you want to compare. The two sets of database definitions to be compared can be from two baselines, two databases, or a database and a baseline. Among the differences that comparisons can identify are text differences in objects such as procedures, packages, or views, and differences in instance parameters.

Create Comparison Versions to Track Definition Changes Over Time

You can use comparison specifications to create multiple versions of a comparison. You can use comparison versions to track changes to two sets of definitions over time.

For example, you can use a comparison specification to create a comparison that identifies the differences between the definitions in the original baseline for your application and those same definitions in your current database. Each time you create a new version of this comparison using the comparison specification, that comparison version identifies the differences between the original definitions at the start of the development cycle and those same definitions at the current time.

You could also use another comparison specification to create a comparison that compares definitions from your most recent (latest) baseline with those same definitions in the previous (2nd latest) baseline. Each time you create a new version of this comparison using the comparison specification, that comparison version identifies the differences in the definitions since the previous baseline.

Generate Reports for Differences Identified in Comparisons

When viewing a comparison in the Comparison Viewer, you can choose **File->List** to generate a report that describes the differences between the two sets of definitions included in the comparison.

You can automate the creation of comparison versions by creating a Change Management Pack command line interface script to capture the comparison and using the facilities of the NT or Solaris operating system to schedule the execution of that script. See [Appendix A, "Command Line Interface Appendix"](#) on page A-1 for more information about the command line interface.

Synchronize Definitions from Comparison Viewer

From Comparison Viewer, you can also start the the Synchronization Wizard and synchronize two sets of definitions. For example, you can use the Synchronization Wizard to synchronize your current database definitions with the definitions

captured in an earlier baseline. Synchronizing definitions to match those in a baseline has the effect of rolling back changes made to your database since the time the baseline was captured.

Reproduce Definitions From One Database or Schema at Another

You can use DB Propagate to reproduce definitions from one schema or database at another. For example, you can reproduce schemas from your development database at your test database. When pre-production time comes, you can then reproduce the test database schemas at your pre-production database, and finally, at production time, reproduce the pre-production schemas at your production database. By default, only definitions are reproduced at the destination database, but when you enable the Copy Table Data propagation option both definitions and data are reproduced. See the online help for more information about propagation options.

Use Change Applications to Make Changes to One or More Definitions

In some cases, you want to make specific changes to one or more definitions. Use DB Quick Change to make specific changes to one definition. Use DB Alter or Plan Editor to make specific changes to multiple definitions.

Whether you are reproducing definitions or making specific changes to applications, you can use Change Management Pack applications to:

- Create a change plan.
- Generate a script from the change plan that include the necessary steps to carry out the change requests in the change plan.
- Generate an impact report that describes the changes the script will make when it executes at a given database.
- Generate an execution log when the script executes.
- Undo or keep the changes made by the script when it executes.

Use Printing, Report, and History Features to Track Definition Changes

You can print impact reports, script summaries, scripts, and execution logs to track the work you do using change plans. See "[Viewing the Impact Report and Script Summary with Plan Editor](#)" on page 3-18 for more information on impact reports, script summaries, and scripts, and "[Executing the Script with Plan Editor](#)" on page 3-20 for more information on execution logs.

You can use Change Manager's history features to see when change plan scripts were generated and executed, baselines were captured, and comparisons were

performed. Change Manager's history features also record which administrator performed these tasks and the status of each task (for example, whether a capture operation succeeded or failed). You can print a report that includes this history information. See "[Viewing History Entries for Tasks](#)" on page 2-11 for more information on using Change Manager's history features.

Using Change Plans to Make Changes

This section provides an overview of the process of creating and modifying object definitions using change plans. It introduces you to important Oracle Change Management Pack concepts and terms.

The process for making changes using change plans involves these phases:

1. Planning and defining changes
2. Evaluating the impact of changes
3. Implementing changes

The following sections describe each of these phases. Note that although change plans can be created with the DB Alter, DB Quick Change, DB Propagate, Synchronization Wizard, and Plan Editor applications, some of these applications may hide various phases of the process from you to reduce the time it takes to make the desired changes.

Planning and Defining Changes

The first phase of using a change plan to create or modify object definitions is to plan and define the changes that you want to make.

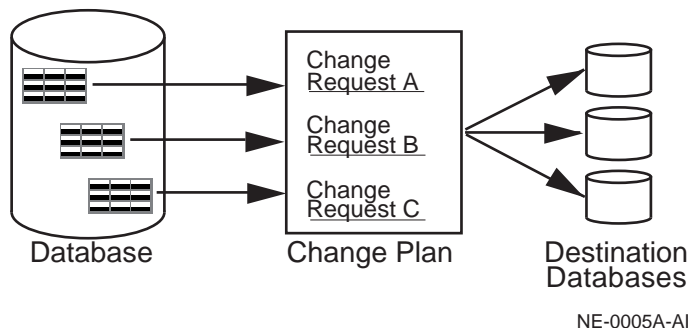
During the planning stage, you plan the changes that you want to make. For example, you may want to make one or more changes to an existing object definition in one or more databases. Or, you may want to reproduce one or more object definitions from one schema or database in another schema or database.

During the definition stage, you use Oracle Change Management Pack to create a change plan and change requests, and you specify one or more destination databases. A change plan is a named container for change requests. See "[Creating a Plan with Plan Editor](#)" on page 3-3 for more information about creating a change plan. Change requests describe to Oracle Change Management Pack the types of changes that you want to make. See "[Understanding Change Requests](#)" on page 3-4 for more information on defining change requests to reproduce or modify object definitions at a destination database. A destination database is a database where

you want to apply the change requests in a change plan. See ["Selecting a Destination Database with Plan Editor"](#) on page 3-14 for more information about selecting a destination database.

[Figure 1-9](#) shows the planning and defining changes phase:

Figure 1-9 Using a Change Plan: Planning and Defining Changes



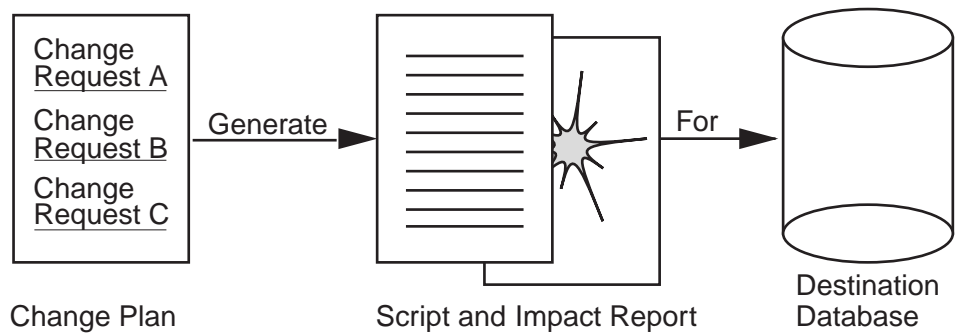
Evaluating the Impact of Changes

After you finish planning and defining the changes, the next phase of the process is to evaluate the impact of the changes that you want to make.

To evaluate the impact of the change requests at a particular database, you use Oracle Change Management Pack applications to generate a script and an impact report for a change plan and that destination database. The impact report explains the changes that will be made by the script when it executes at the destination database. It also describes any change requests that cannot be applied at the destination database, and explains how to modify the destination database or change plan so that those changes can be made. See ["Viewing the Impact Report and Script Summary with Plan Editor"](#) on page 3-18 for more information about the impact report.

The script is used to implement the desired changes at the destination database. See ["Generating a Script with Plan Editor"](#) on page 3-15 for more information about generating a script for a destination database.

[Figure 1-10](#) shows the evaluating the impact of changes phase:

Figure 1–10 Using a Change Plan: Evaluating the Impact of Changes

NE-0006A-AI

Implementing the Changes

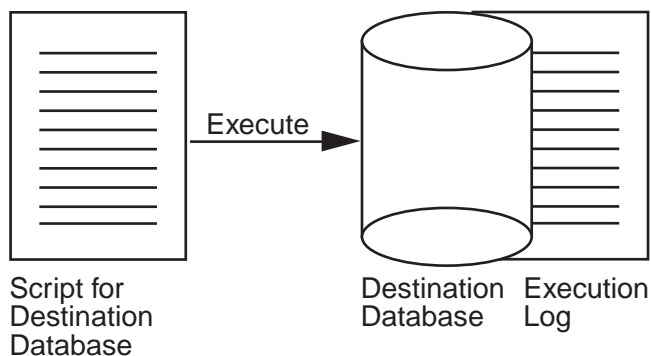
The last phase of the process is to implement the changes.

To implement the change requests in a change plan at a destination database, you use Oracle Change Management Pack to execute the script at the destination database. See ["Executing the Script with Plan Editor"](#) on page 3-20 for more information about executing a script at a destination database.

During or after script execution, you can examine the execution log, which displays messages, including the status of the script execution ("Script execution succeeded" or "Script execution failed."). See ["Dealing with Script Execution Errors"](#) on page 3-22 for more information on understanding execution log error messages.

You are also able to keep or undo the changes made by the script at the destination database. See ["Executing the Script with Plan Editor"](#) on page 3-20 for more information about keeping or undoing changes made by a change plan script.

[Figure 1–11](#) shows the implementing the changes phase:

Figure 1–11 Using a Change Plan: Implementing the Changes

NE-0007A-AI

Learning More About Making Changes Using Change Plans

The DB Alter, DB Quick Change, DB Propagate, and Synchronization Wizard change applications and Plan Editor use many of the same components to create change plans, scripts, impact reports, and execution logs. Therefore, after you have used a components in one of the applications, you will know how to use it in another.

The DB Alter, DB Quick Change, DB Propagate, and Synchronization Wizard change applications have page-oriented interfaces that guide you step-by-step through the process of specifying the changes you want to make and making those changes. Each of these applications is designed to create a change plan that can be used to make a particular type of change.

With Plan Editor, you are not constrained to creating a change plan that makes only one type of change. Plan Editor is the most flexible change application. It allows you to make any of the changes that you can make with the other change applications.

Because Plan Editor can create a change plan that can carry out any type of change request, it does not have a page-oriented interface that guides you through the steps of making a particular type of change. Therefore, you need to learn the process for making changes using Plan Editor.

For detailed information on the process for making changes using Plan Editor, see [Chapter 3, "Using Plan Editor"](#) on page 3-1.

Using Help

Online help is provided for all of the Oracle Change Management Pack applications. You can display the Contents page of the Oracle Change Management Pack help system, or you can display a help topic for your current context in an application.

Displaying the Contents Page for Help

Depending on the application that you are using, you can display the Contents page for the Oracle Change Management Pack online help by using one of the following methods:

- Press F1. If the Contents page for the help system does not display, click the Navigator button on the Help Topic Window toolbar.
- Click the ? on the application toolbar. This applies to Change Manager, Plan Editor, Baseline Viewer, and Comparison Viewer only.
- Choose **Help->Contents**. This applies to Change Manager, Plan Editor, Baseline Viewer, and Comparison Viewer only.

Use the help system's Contents page or Help Search page to locate help topics of interest. To access the Help Search page, choose **Tools->Search** on the Help Navigator window.

Displaying a Help Topic for Your Current Context in an Application

You can display context-sensitive help in several ways, depending on your context in an application.

- If a Help button exists, click it.
- If a Help button does not exist, click the object that you are interested in, then press F1.
- Some wizard pages for the Oracle Change Management Pack applications also display property pages. To access help for the wizard page, click the Help button at the bottom left corner of the page. To access help for a property page, click a field in the property page, then click the Help button beneath the property page. If a Help button does not exist beneath the property page, click a field in the property page, then press F1.

Finding a Particular Type of Help Topic

In addition to providing context-sensitive help, the Oracle Change Management Pack help system contains conceptual topics, task topics, and a glossary. To locate each type of topic:

- Help topics whose titles begin with words such as "About," "Introduction," or "Overview" are conceptual topics. These topics usually provide overview information about Oracle Change Management Pack applications or conceptual information that you should be aware of before performing a particular task. They do not provide step-by-step instructions for accomplishing a task. The Conceptual Topics book on the help system's Contents page includes all of the conceptual topics in the Oracle Change Management Pack help system.
- Help topics whose title begin with the word "To" are task topics. These topics provide step-by-step instructions for accomplishing a task. They do not provide conceptual information that you should be aware of before performing the task. The Task Topics book on the help system's Contents page includes all of the task topics in the Oracle Change Management Pack help system.
- The Oracle Change Management Pack glossary is available by accessing the Glossary book on the help system's Contents page.

Using Change Manager

The Change Manager application is the Oracle Change Management Pack central interface. Change Manager gives you direct or indirect access to all of the Oracle Change Management Pack operations (in cases where Change Manager cannot directly perform an operation, such as capturing or comparing object definitions, Change Manager starts the application that performs that operation).

Running Change Manager Connected to an Oracle Management Server or Standalone

You can run Change Manager connected to an Oracle Management Server or not connected to an Oracle Management Server. When you run Change Manager without connecting to an Oracle Management Server, you are running Change Manager standalone.

When you run Change Manager connected to an Oracle Management Server:

- Change Manager provides access to all the Oracle Change Management Pack objects owned by all users in the Oracle Management Server repository. Objects are ordered by owner in the Change Manager navigator tree.
- Each user has read and write access to the objects that he or she owns, and has read-only access to objects owned by other users.
- Other Change Management Pack applications started from Change Manager also run connected to the Oracle Management Server.
- You can submit change plan scripts as jobs using the Oracle Enterprise Manager job system.

When you run Change Manager standalone:

- Change Manager provides access to the Oracle Change Management Pack objects in your standalone repository. Because the standalone repository contains only objects that you created, the Change Manager navigator tree displays only your Change Management Pack objects.
- You have full access to your objects, but no access to any other objects.
- Other Change Management Pack applications started from Change Manager also run standalone.
- You cannot submit change plan scripts as jobs using the Oracle Enterprise Manager job system, but you can execute scripts interactively.

Creating a Standalone Repository for Change Management Pack Applications

A standalone repository is required to store Change Management Pack objects created during a standalone Change Management Pack session. Read and follow the steps to perform before creating a standalone repository in the *Oracle Enterprise Manager Configuration Guide*.

The next time you start Change Manager standalone, a repository login dialog box prompts you for the Username, Password, and Service credentials of the user that owns the schema to be used for your standalone repository. When you provide the credentials for that user, the standalone repository tables needed by the Oracle Change Management Pack applications will be created for you in that user's schema.

Changing Your Standalone Change Management Pack Repository

At some point after creating a standalone repository for Change Management Pack objects created during your Change Management Pack sessions, you might decide that you want to use a different standalone repository.

To start using a different standalone repository, you must:

1. Make sure that you have followed the steps in the *Oracle Enterprise Manager Configuration Guide* that are a prerequisite for creating the new standalone repository.
2. In the `ORACLE_HOME/sysman/config/ClientConfig.properties` file (where `ORACLE_HOME` is the directory in which Oracle Enterprise Manager is installed), delete the lines that contain the following parameters:

- OEM.REPOSITORY.USER
- OEM.REPOSITORY.PASSWORD
- OEM.REPOSITORY.DATABASE

Removing these lines from ClientConfig.properties removes information about the current standalone repository that is being used for standalone Change Management Pack objects.

3. The next time you start Change Manager standalone, you are prompted for repository credentials for your standalone repository. Specify the Username, Password, and Service credentials of the user that owns the schema to be used for your standalone repository. The Username, Password, and Service information that you specify for the new standalone repository is stored in the ClientConfig.properties file.

Change Manager uses the information in the ClientConfig.properties file to determine the standalone repository in which to store information from your standalone Change Management session.

Starting Change Manager

You can start the Change Manager application in any of the following ways:

- From the Oracle Enterprise Manager Console, use any of these methods:
 - Select a database in the navigator panel, click the right mouse button, point to the **Related Tools** menu option, then click **Change Manager**.
 - On the **Tools** menu, point to **Change Management Pack**, then click **Change Manager**.
 - Click the Change Management Pack drawer, then click the Change Manager icon.

When Change Manager is started from the Console using any of these methods, the mode in which the Console is running (connected to the Oracle Management Server or standalone) determines the mode in which Change Manager will run.

- On UNIX or in a DOS window on an NT machine, set default to the ORACLE_HOME/bin directory (where ORACLE_HOME is the directory in which Oracle Enterprise Manager is installed), then enter the following in lowercase characters at the command line prompt:

```
oemapp ocm
```

When you start Change Manager using this method, you are prompted whether you want to run Change Manager connected to the Oracle Management Server or standalone.

- On the Windows **Start** menu, point to **Programs**, then to the Oracle Home where Oracle Enterprise Manager is installed, then to **Change Management Pack**, then click **Change Manager**.

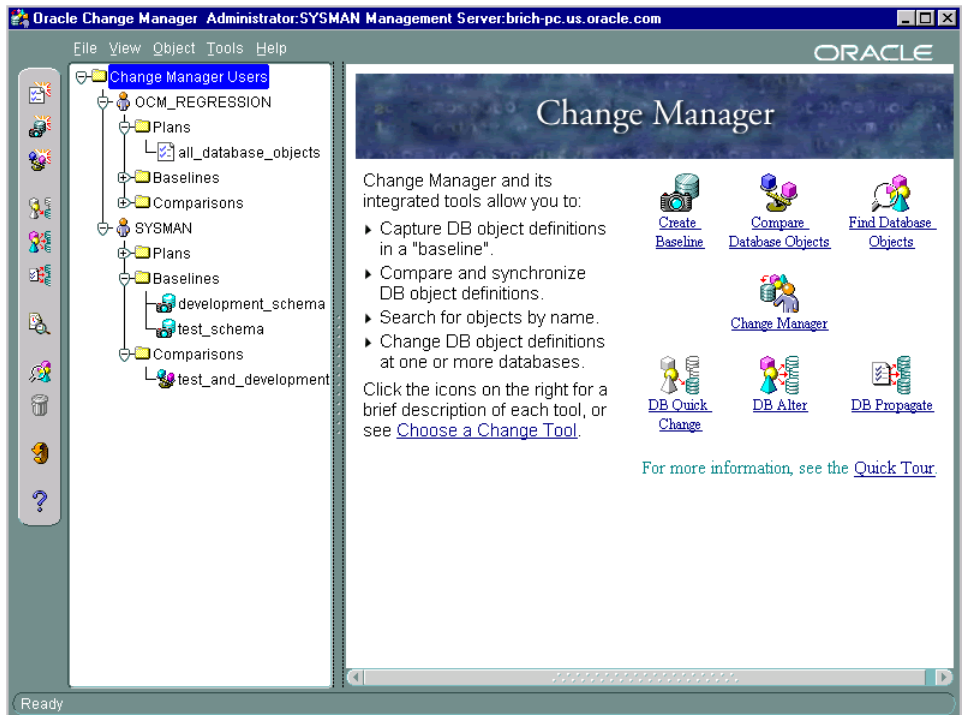
When you start Change Manager using this method, you are prompted whether you want to run Change Manager connected to the Oracle Management Server or standalone.

See "[Starting Change Management Pack Applications from Change Manager](#)" on page 2-6 for more information on starting the other Change Management Pack applications from Change Manager.

Using the Change Manager Main Window

The Change Manager main window (see [Figure 2-1](#)) has a navigator tree on the left and a detail view on the right.

Figure 2–1 Change Manager Navigator Tree and Detail View



In **Figure 2–1**, Change Manager is running connected to an Oracle Management Server. The navigator tree displays the following Oracle Change Management Pack objects, organized by owner:

- change plans
- baselines
- comparisons

The tree contains the Plans, Baselines, and Comparisons folders, even if objects of these types have not been created yet.

If Change Manager is running standalone, the tree displays only the Change Management Pack objects belonging to the owner of the standalone repository being used.

If the Change Manager navigator tree contains no objects, this means that no change plans, baselines, or comparisons have been created in the repository. When you create these objects, they appear in the Change Manager tree.

When you select a Change Management Pack object or object folder and click the right mouse button, a context menu is displayed that lists the operations that can be performed in the current context. If you select an object that you do not own, some of the operations on the context menu are unavailable (for example, you cannot remove or modify an object that you do not own, because you have read-only access to objects owned by other users). On the context menu, choose any available operation that you are interested in performing.

When the Change Manager Users folder (the root of the Change Manager tree) is selected (as in [Figure 2-1](#)), the detail view to the right of the Change Manager tree provides information about Oracle Change Management Pack applications and the operations that they can perform. Click on an application's icon in the detail view to learn more about the operations that the application can perform.

When other folders or objects are selected in the Change Manager tree, the detail view provides related information about the selected folder or object.

Note: Change Manager provides right mouse button support for some operations.

After you select an object in the Change Manager navigator, click the right mouse button to display a menu of options. Any menu options that are not appropriate for the given object are unavailable from the menu.

Starting Change Management Pack Applications from Change Manager

You start the other Oracle Change Management Pack applications and viewers from Change Manager, as follows:

- DB Alter
Choose **Tools->DB Alter**.
- Create Baseline
Choose **Object->Create Baseline**.
- Compare Database Objects
Choose **Object->Compare Database Objects**.

- DB Propagate
Choose **Tools->DB Propagate**.
- DB Quick Change
Choose **Tools->DB Quick Change**.
- Plan Editor
Choose **Object->Create Change Plan**, then in the Create Change Plan Options dialog box, click **Create Change Plan Manually**.
- Baseline Viewer
Expand the Change Manager folder, user folder, and Baselines folder in the Change Manager tree, select a baseline, then choose **Object->View Baseline**.
- Comparison Viewer
Expand the Change Manager folder, user folder, and Comparisons folder in the Change Manager tree, select a comparison, then choose **Object->View Comparison**.

You can also start a Change Management Pack application by clicking the Change Manager toolbar icon for that application.

When you start a Change Management Pack application from Change Manager, the mode in which Change Manager is running (connected to the Oracle Management Server or standalone) determines the mode in which the other Change Management Pack application will run.

Working with Baselines

[Table 2-1](#) describes how to use Change Manager to perform different types of baseline operations.

Table 2-1 Performing Baseline Operations

Operation	Steps for Performing the Operation
Create a new baseline	Choose Object->Create Baseline and follow the steps in the Create Baseline application.
View a baseline	Select the baseline in the tree, then choose Object->View Baseline .
Delete a baseline	Select the baseline in the tree, then choose Object->Remove . This deletes all the versions of the baseline.

Table 2–1 Performing Baseline Operations (Cont.)

Operation	Steps for Performing the Operation
Export a baseline to a file	Choose Tools->Export Plans/Baselines/Comparisons , then specify the baseline to export and the name of the export file.
Import a baseline from a file	Choose Tools->Import Plans/Baselines/Comparisons , then specify the name of the export file and the baseline to import.
Create a new version of a baseline	Select the baseline in the tree, then choose Object->Recapture .
View a previous version of a baseline	Select the baseline in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to view, then click View .
Delete a previous version of a baseline	Select the baseline in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to delete, then click Remove .
View history entries for baseline tasks	Choose View->Show History . In the History dialog box, view history entries for completed baseline tasks on the History page.

For more information on working with baselines, see:

- ["Viewing History Entries for Tasks"](#) on page 2-11 for more information on viewing history entries for baseline tasks.
- ["Working With a Particular Version of a Change Plan, Baseline, or Comparison"](#) on page 2-13 for more information on working with particular versions of baselines.
- ["Exporting and Importing Change Plans, Baselines, and Comparisons"](#) on page 2-14 for more information on exporting and importing baselines.

Working with Comparisons

[Table 2–2](#) describes how to use Change Manager to perform different types of comparison operations.

Table 2–2 Performing Comparison Operations

Operation	Steps for Performing the Operation
Create a new comparison	Choose Object->Compare Database Objects and follow the steps in the Compare Database Objects application.

Table 2–2 Performing Comparison Operations (Cont.)

Operation	Steps for Performing the Operation
View a comparison	Select the comparison in the tree, then choose Object->View Comparison .
Delete a comparison	Select the comparison in the tree, then choose Object->Remove . This deletes all the versions of the comparison.
Export a comparison to a file	Choose Tools->Export Plans/Baselines/Comparisons , then specify the comparison to export and the name of the export file.
Import a comparison from a file	Choose Tools->Import Plans/Baselines/Comparisons , then specify the name of the export file and the comparison to import.
Create a new version of a comparison	Select the comparison in the tree, then choose Object->Repeat Comparison .
View a previous version of a comparison	Select the comparison in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to view, then click View .
Delete a previous version of a comparison	Select the comparison in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to delete, then click Remove .
View history entries for comparison tasks	Choose View->Show History . In the History dialog box, view history entries for completed comparison tasks on the History page.

For more information on working with comparisons, see:

- ["Viewing History Entries for Tasks"](#) on page 2-11 for more information on viewing history entries for comparison tasks.
- ["Working With a Particular Version of a Change Plan, Baseline, or Comparison"](#) on page 2-13 for more information on working with particular versions of comparisons.
- ["Exporting and Importing Change Plans, Baselines, and Comparisons"](#) on page 2-14 for more information on exporting and importing comparisons.

Working with Change Plans

Table 2–3 describes how to use Change Manager to perform different types of change plan operations.

Table 2–3 Performing Change Plan Operations

Operation	Steps for Performing the Operation
Create a new change plan	Choose Object->Create Change Plan . Then in the Create Change Plan Options dialog box, select the name of the application that you want to use to create the change plan (the Create Manually option lets you use Plan Editor to create the plan). See Chapter 3, "Using Plan Editor" for more information on using Plan Editor to create a change plan.
View and modify a change plan	Select the change plan in the tree, then choose Object->Edit Change Plan . This lets you view and modify the change plan using Plan Editor. See Chapter 3, "Using Plan Editor" for more information on using Plan Editor to edit and modify a change plan.
Delete a change plan	Select the change plan in the tree, then choose Object->Remove . This deletes all versions of the change plan.
Rename a change plan	Select the change plan in the tree, then choose Object->Edit Change Plan . On the General page for the plan in Plan Editor, edit the name of the plan, then choose File->Refresh . This updates the plan name in the Plan Editor tree. Choose View->Refresh in Change Manager to see the plan's new name displayed in the Change Manager tree.
Export a change plan to a file	Choose Tools->Export Plans/Baselines/Comparisons , then specify the change plan to export and the name of the export file.
Import a change plan from a file	Choose Tools->Import Plans/Baselines/Comparisons , then specify the name of the export file and the change plan to import.
Create a new version of a change plan	None. Oracle Change Management Pack automatically creates a new version of a change plan when necessary to ensure that the history features for the change plan work properly.
View a previous version of a change plan	Select the change plan in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to view, then click View .
Delete a previous version of a change plan	Select the change plan in the tree, then choose Object->Show Versions . In the Versions dialog box, click the version you want to delete, then click Remove .

Table 2–3 Performing Change Plan Operations (Cont.)

Operation	Steps for Performing the Operation
View history entries for change plan tasks	Choose View->Show History . In the History dialog box, view history entries for pending plan tasks on the Pending page and view history entries for completed plan tasks on the History page.

For more information on working with change plans, see:

- ["Viewing History Entries for Tasks"](#) on page 2-11 for more information on viewing history entries for change plan tasks.
- ["Working With a Particular Version of a Change Plan, Baseline, or Comparison"](#) on page 2-13 for more information on working with particular versions of change plans.
- ["Exporting and Importing Change Plans, Baselines, and Comparisons"](#) on page 2-14 for more information on exporting and importing change plans.

Viewing History Entries for Tasks

You can view history entries that show the status of the following Oracle Change Management Pack tasks:

- Script generation
- Script execution
- Capturing baselines
- Performing comparisons

By viewing the history entries for pending tasks and completed tasks, you can determine when each Oracle Change Management Pack task was performed and its completion status.

To view the history entries for tasks in the current Oracle Change Management Pack repository, on the Change Manager **View** menu, choose the **Show History** option, which displays the History dialog box. The History dialog box allows you to view both pending tasks and completed tasks.

Viewing and Completing Pending Tasks

Click the Pending tab of the History dialog box to view pending tasks. Pending tasks are script executions that need to be kept or undone. See ["Executing the Script with Plan Editor"](#) on page 3-20 for more information on keeping or undoing the changes made by a change plan script.

The history entries on the Pending page of the History dialog box enable you to keep track of the Oracle Change Management Pack pending tasks, which need further attention. Each row on the Pending page is a history entry for one pending task.

When you complete a pending task and refresh the History dialog box, the entry is moved to the History page. You also have the option of manually moving an entry for a pending task to the History page.

On the Pending page, you can perform the operations shown in [Table 2-4](#):

Table 2-4 Operations That Can be Performed on Pending Tasks

Operation	Steps to Perform the Operation
Complete a pending task	Select a pending task and click the GoTo button. On the Execution Log page, click Keep or Undo to keep or undo the changes made by the script execution.
Update the pending tasks display	Click the Refresh button. Any new pending tasks are displayed. Any previously pending tasks that have been completed are removed from the Pending page and displayed on the History page.
Manually move a pending task to the History page	Select a pending task and click the Move to History button. This moves the pending task to the History page, where you have the option of completing it later.
Save history entries to a file	Click the Save List button. The Save List dialog box lets you save some or all of the pending task history entries in a file. You have the option of viewing and printing the information to be included in the file before saving the file.
Sort history entries	Click a column heading on the Pending page. The first time you click a heading, the history entries are sorted in ascending order, using the values in that column. If you click again on the same heading, the history entries are sorted in descending order, using the values in that column.

You are notified of any pending tasks on the Pending page and are given the opportunity to complete them each time you exit Change Manager.

Viewing Completed Tasks

Click the History tab of the History dialog box to view completed tasks.

The history entries on the History page of the History dialog box enable you to track the Oracle Change Management Pack tasks that have been completed or that have been manually moved from the Pending page to the History page. Each row on the History page is a history entry for one completed task or a pending task that was manually moved from the Pending page. The status for a particular history entry tells you whether the task completed successfully or not.

On the History page, you can perform the operations shown in [Table 2-5](#):

Table 2-5 Operations That Can be Performed on Completed or Manually Moved Tasks

Operation	Steps to Perform the Operation
Remove a task	Select a task and click Remove .
Update the History page display	Click the Refresh button. Any new completed tasks are displayed.
View a task	Select a task and click GoTo . The appropriate Oracle Change Management Pack window is opened in the correct context for viewing the task.
Complete a pending task that was manually moved to the History page	Select the pending task and click GoTo . Click Keep or Undo on the Execution Log page to keep or undo the changes made by the script execution.
Save history entries to a file	Click the Save List button. The Save List dialog box lets you save some or all of the task history entries in a file. You have the option of viewing and printing the information to be included in the file before saving the file.
Sort history entries	Click a column heading on the History page. The first time you click a heading, the history entries are sorted in ascending order, using the values in that column. If you click again on the same heading, the history entries are sorted in descending order, using the values in that column.

Working With a Particular Version of a Change Plan, Baseline, or Comparison

After you create a baseline specification, you can use the specification to generate multiple baselines over time. Similarly, after you create a comparison specification, you can use it to generate multiple comparisons over time. New versions of a

change plan are created automatically when necessary to ensure that the history features for change plans work properly.

By default, when you select a change plan, baseline, or comparison in Change Manager, any operations you perform are carried out on the latest version of that object. In some cases, however, you may want to view or remove an earlier version of the object. To do so, after you select the object in Change Manager, on the **Object** menu, click **Show Versions**. This displays the Versions dialog box. In the Versions dialog box, select the version of the object that you are interested in, then click **View** to view the selected version of the object in the appropriate Oracle Change Management Pack window or click **Remove** to remove the selected version of the object from the Oracle Enterprise Manager repository.

Note: When you remove a version of a change plan, baseline, or comparison, the version numbers for the remaining versions of that object remain the same.

Exporting and Importing Change Plans, Baselines, and Comparisons

You can export and import change plans, baselines, and comparisons using Change Manager. After you export one or more of these objects to a file, you can then import the objects into another repository, which copies the objects from one repository to another.

To export change plans, baselines, or comparisons:

1. In Change Manager, choose **Tools->Export Plans/Baselines/Comparisons**.
2. In the Export Plans/Baselines/Comparisons dialog box, select the object or objects to export, then click **Export**.
3. In the Export Plans/Baselines/Comparisons to which file? dialog box, select a file into which the selected objects will be exported, then click **Save**.

To import change plans, baselines, or comparisons:

1. In Change Manager, choose **Tools->Import Plans/Baselines/Comparisons**.
2. In the Import Plans/Baselines/Comparisons from which file? dialog box, identify the file that contains the object or objects that you want to import, then click **Open**.
3. In the Import Plans/Baselines/Comparisons dialog box, select the object or objects that you want to import, then click **Import**. If the import utility discovers that any of the objects being imported has the same name as an

existing object in the current repository, you will be prompted to either rename the object being imported or to cancel the import operation for that object.

Using Plan Editor

The Plan Editor application allows you to modify and/or reproduce database object definitions at one or more databases.

With Plan Editor, you create or modify a single change plan and populate it with change requests. To deploy the plan, you specify one or more destination databases, generate a script to apply the plan's change requests at each database, then execute the scripts.

You can also create change plans with DB Alter, DB Quick Change, DB Propagate, and the Synchronization Wizard. However, these change applications differ from Plan Editor because each of them is designed to create a change plan with specific types of change requests that make specific types of changes.

Plan Editor is a more flexible change plan tool. You can use Plan Editor to create and modify a change plan that includes any type of change request and to make a wider variety of changes. Plan Editor can edit plans that were created earlier using the other change applications and can generate and execute scripts for those plans.

The other change applications have page-oriented interfaces that guide you step-by-step through the process of specifying the changes the tool is designed to make and making those changes. Because you can make different types of changes using Plan Editor, its interface is not page-oriented and it does not guide you step-by-step. You must learn the process of using Plan Editor to create and modify change plans that make different types of changes.

Note: This chapter explains in detail the steps for modifying and creating object definitions using Plan Editor. In the process of describing these steps, the chapter provides information about the following, which are available in other change applications besides Plan Editor:

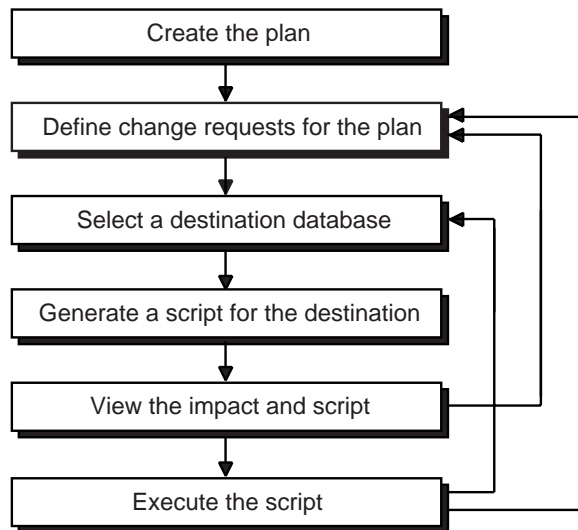
- change requests (directives, scoped directives, exemplars, and modified exemplars)
- operations (change plan creation, destination database selection, script generation, and script execution)
- features (viewing grants, viewing dependent and dependency objects, and using propagation options)

Therefore, when you read this chapter, you learn how to use Plan Editor effectively, and you learn about concepts, operations, and features that will enable you to use the other change applications more effectively.

Modifying and Creating Object Definitions with Plan Editor

This section describes how to start the Plan Editor application and use it to create a change plan, which is used to modify and create object definitions.

[Figure 3–1](#) shows the steps for creating a change plan. The steps in the figure are described in detail in the remainder of this chapter.

Figure 3–1 *Creating a Change Plan with Plan Editor*

NU-3708A-AI

Note: Plan Editor provides right mouse button support for some operations.

After you select an object in the Plan Editor navigator, click the right mouse button to display a menu of options. Any menu options that are not appropriate for the selected object are unavailable from the menu.

Creating a Plan with Plan Editor

To create a new plan using Plan Editor, follow these steps:

1. Start Change Manager. See "[Starting Change Manager](#)" on page 2-3 for information on the different ways to start Change Manager.
2. On the Change Manager **Object** menu, click **Create Change Plan**.
3. In the Create Change Plan Options dialog box, click **Manual Creation** and then **OK**, which displays the Create Plan dialog box.

4. Create a plan by following these steps, which are also shown in [Figure 3-1](#) on page 3-3:
 - a. On the General page of the Create Plan dialog box:
 - * Supply a unique name for the plan. Oracle Change Management Pack allows plan names and baseline names of up to 50 characters in length. Any character, including blank, is allowed. However, to avoid confusion, it is recommended that you do not use leading or trailing blanks in a plan name or baseline name.
 - * Choose the plan's source database (the database used to create the plan's change requests) from the list of databases. The list of databases are the databases known to (discovered by) the Oracle Enterprise Manager Console (or the list of databases discovered in the standalone Console, if you are running Plan Manager standalone).
 - * Provide a description of the plan (optional).
 - b. Click **Create** to create the plan. This launches Plan Editor for the newly-created plan, which is empty when it is created.

Understanding Change Requests

After Plan Editor creates the empty plan, you need to define one or more change requests to add to the plan. Depending on what you want the plan to accomplish, you will define directives, scoped directives, exemplars, modified exemplars, or some combination of these types of change requests for the plan.

The rest of this section provides conceptual information about directives, scoped directives, exemplars, and modified exemplars.

Understanding Directives

A directive is a set of changes you specify for an existing, named object definition. A directive can be thought of as a “Super Alter” statement.

A directive for an object definition tells an Oracle Change Management Pack application, “Make these specific changes to the object definition.” Suppose, for example, that you are defining a directive for a table named `Table_1`. Some of the change requests that you could specify in a directive for table `Table_1` include:

- Adding column `REGION_ID` with data type of `NUMBER(3)`.
- Renaming the column `REGNAME` to `AREA_NAME`.
- Changing the tablespace for the table to tablespace `TABLESPACE_1`.

You create a directive for an object definition by selecting an object definition to be modified, then specifying the changes to the object definition's attributes on property sheets similar to those used in the Oracle Enterprise Manager Console.

A directive has the scope of a single object definition. That is, the changes specified in the directive are applied to a single object definition when the change plan script executes at a destination database.

Understanding Scoped Directives

You can also extend the scope of a directive so that the changes specified in the directive are applied to multiple object definitions that you specify. A directive with an extended scope is called a scoped directive. The changes specified in a scoped directive are applied to multiple object definitions when the change plan script executes at a destination database.

Understanding Exemplars

An exemplar is a complete object definition to be reproduced, either by creating a new object definition or by modifying an existing object definition of the same name and object type. An exemplar can be thought of as the example of what you want to reproduce.

An exemplar tells an Oracle Change Management Pack application, "Reproduce this object definition. If an object definition of the same name and type already exists at the destination database, make whatever changes are necessary to that object definition so that it matches this object definition. If the object definition does not exist at the destination database, create an object definition that matches this object definition." When you define an exemplar, you can also include grants information for the exemplar.

Understanding Modified Exemplars

You can also edit the attributes of an exemplar definition before propagating it. For example, you can modify the storage parameters for exemplar definitions before propagating the definitions from a production database to a test database. After the propagation operation completes, the propagated definitions will already have the desired storage parameters for the new destination. An exemplar with attributes that have been changed since the exemplar was created is referred to as a modified exemplar. See "[Creating a Modified Exemplar with Plan Editor](#)" on page 3-13 for more information on using Plan Editor to create a modified exemplar.

Table 3-1 summarizes the types of change requests that can be created with Oracle Change Management Pack applications.

Table 3-1 Change Requests Created with Oracle Change Management Pack Applications

Application	Type of Change Requests Created
Synchronization Wizard	exemplars
DB Propagate	exemplars, modified exemplars, ¹ or both
DB Quick Change	directives
DB Alter	directives, scoped directives, or both
Plan Editor	directives, scoped directives, exemplars, and modified exemplars, or any combination of these

¹ Modified exemplars can be created with either DB Propagate or Plan Editor, but Plan Editor offers more options for applying modified exemplars in change plan scripts at destination databases. See ["Generating a Script with Plan Editor"](#) on page 3-15 for more information.

Some Oracle Change Management Pack applications (such as DB Alter and DB Propagate) make it evident when you are creating change requests because these applications prompt you to add directives or exemplars to the change plan. Other applications (such as the Synchronization Wizard and DB Quick Change) prompt you to select object definitions to copy or modify, but do not make it evident that the selected definitions are directives or exemplars that are being added to the change plan.

Note: Plan Editor is the only application that can create change plans with all the change request types. Use Plan Editor to create change plans with any combination of the change request types.

Defining Change Requests with Plan Editor

You have created an empty plan using Plan Editor. Now you need to define one or more change requests to add to the plan. Depending on what you want the plan to accomplish, you will define directives, scoped directives, exemplars, modified exemplars, or some combination of these types of change requests for the plan.

The rest of this section provides detailed information about how to create directives, scoped directives, exemplars, and modified exemplars using Plan Editor.

Defining a Directive with Plan Editor

To specify the modifications to be made to an existing object definition at the destination database, follow these steps to define a directive for that object definition in the source database:

1. In Plan Editor, choose **Plan->New Change Requests**.
2. Expand the **Source Database** tree in the New Change Requests dialog box, then select one or more object definitions for which you want to create directives. Definitions that are already in the plan are shown in gray and cannot be selected.
3. Click the **Directive** button. Each selected object definition in the **Source Database** tree turns gray when the directive for the object definition is created. This step also causes the selected directives to be displayed in the **Change Plan** tree.
4. Click **Close** or leave the New Change Requests dialog box displayed if you think you would like to add other change requests to the plan later.
5. In the Plan Editor tree, expand the Change Requests folder under the new plan to display folders for each object type for which change requests have been defined.
6. Expand an object type folder to view the change requests that have been defined for that object type. Directives are marked with both the directive icon and an object type icon. The directive icon appears first, followed by the object type icon, then the name of the object. The directive icon is shown in [Figure 3-2](#).
7. Click the name of a directive in the Plan Editor tree. This displays the General page for the directive in Plan Editor's detail view.
8. On the General page, click the **Edit Directive** button. This displays the Edit dialog box for the directive. Property sheets for the object are displayed in the Edit dialog box. You use the property pages for the directive to specify the changes that you want to make to the directive object definition.
9. On the property pages, specify the changes you want to make to the object definition. Click **OK** to accept the changes you have specified.

Figure 3-2 The Directive Icon



Note: A directive describes the changes you want to make to an object definition. Therefore, the meaningful part of a directive is the changes that you have specified. You can view these changes by clicking the directive in the Plan Editor tree and viewing the list of changes on the directive's General page.

After you have created a directive for an object, you can make further edits to the directive if the object still exists in the change plan's source database. When you make further edits to a directive, the object's definition is loaded from the source database, and all of the changes that are specified for the directive are displayed on the directive's General page.

Defining a Scoped Directive with Plan Editor

By default, a directive contains the changes to be applied to a single object definition. However, it is possible to extend the scope of a directive so that the changes specified in the directive are applied to multiple object definitions that you specify. A directive with an extended scope is called a scoped directive.

Scoped directives can be very powerful. For example, suppose that you want to move several tables to a new tablespace without creating a separate directive for each table you want to move. Instead, you can create a directive for one of the tables, and in that directive specify the name of the new tablespace for the table. Then you can edit the scope specification for that directive, making sure that the search criteria identifies the other tables that you want to move to the new tablespace. When the script is generated, the scoped directive's instructions are applied to each object that matches the search criteria. When the script executes, it carries out the scoped directive's instructions on each matched object. In this example, all the tables that match the scope directive's search criteria are moved to the new tablespace that is specified in the scoped directive.

To create a scoped directive, follow these steps:

1. Make sure that the directive whose scope specification you want to modify already has one or more changes specified for it. If you followed the instructions in "[Defining a Directive with Plan Editor](#)" on page 3-7 to create the directive, you have already completed this step.
2. In the Plan Editor tree, click the directive for which you want to edit the scope specification.
3. On the General page for the directive, click the **Edit Scope** button.

4. In the Edit Directive Scope dialog box, specify a set of search criteria to identify the database object definitions to which the changes specified in the scoped directive will be applied.

The Edit Directive Scope dialog box contains the following fields:

- **Object Type**

Displays the object type of the directive object. This field is non-editable.

- **Schemas**

If the directive object is a non-schema object, you cannot modify this field.

If the directive object is a schema object, you can modify this field. To include any schema name in the search criteria, click **Any Schema**. To include any schema name except the SYS and SYSTEM schemas, click **Exclude SYS, SYSTEM** after clicking **Any Schema**.

To include specific schemas or wild card patterns in the search criteria, click **Add**. The Select Schemas dialog box is displayed. Use the Select Schemas dialog box to add specific schemas or wild card patterns to the search criteria. If you specify wild card patterns, any schema name that matches the pattern will be included in the search criteria.

To remove specific schemas or wild card patterns from the search criteria, select the item in the **Schemas** list, then click **Remove**.

- **Search For**

To include any object name in the search criteria, click **Any Object Name**.

To include specific object names or wild card patterns in the search criteria, click **Specified Object Names** and then click **Add**. The Select Names dialog box is displayed. Use the Select Names dialog box to add specific object names or wild card patterns to the search criteria. If you specify wild card patterns, any object name that matches the pattern will be included in the search criteria.

To remove specific schema names or wild card patterns from the search criteria, select the item in the **Search For** list, then click **Remove**.

See the online help for more information about specifying wild card patterns.

5. Click **OK** to confirm the search criteria for the scoped directive. A Scope Specification box that includes the search criteria that you chose appears on the General page for the directive. Also, in the Plan Editor tree, the directive icon changes to the scoped directive icon.

Scoped directives are marked with both the scoped directive icon and an object type icon. The scoped directive icon appears first, followed by the object type icon, then the name of the object. The scoped directive icon is shown in [Figure 3-3](#).

Figure 3-3 The Scoped Directive Icon



A change plan can have only one change request for an object in a destination database. For example, a plan cannot contain both a directive and a scoped directive for the same object in a destination database. If a change plan contains multiple change requests for the same object, this problem will be identified in the impact report.

Refer to the online help for more information on scoped directives.

Defining an Exemplar with Plan Editor

To reproduce an existing object definition at a destination database, you create an exemplar for that object definition in the source database. Later, when the script generated from the change plan is executed against the destination database, one of the following three actions is performed for each exemplar in the change plan:

- If an object definition of the same name and type as the exemplar does not exist at the destination database, the script creates the object at the destination database.
- If an object definition of the same name and type as the exemplar exists at the destination database but the object definition is different than the exemplar's definition, the script makes the necessary changes to the object definition so that it matches the exemplar's definition.
- If an object definition of the same name and type as the exemplar exists at the destination database and the object definition exactly matches the exemplar's definition, no changes are made to the object definition at the destination database.

To reproduce an existing object definition at a destination database, follow these steps to create an exemplar for that object definition in the source database:

1. On the Plan Editor **Plan** menu, click **New Change Requests**.
2. Expand the **Source Database** tree in the New Change Requests dialog box, then select one or more object definitions for which you want to create exemplars.

Definitions that are already in the plan are shown in gray and cannot be selected.

3. Click the **Exemplar** button. Each selected object definition in the **Source Database** tree turns gray when the exemplar for the object definition is created. This step also causes the selected exemplars to be displayed in the **Change Plan** tree.
4. Click **Close** or leave the New Change Requests dialog box displayed if you think you would like to add other change requests to the plan later.
5. In the Plan Editor tree, expand the Change Requests folder under the new plan, which displays folders for each object type for which change requests have been defined.
6. Expand an object type folder to view the change requests that have been defined for that object type. Exemplars are marked with both an object type icon and with the exemplar icon. The exemplar icon appears first, followed by the object type icon, then the name of the object. The exemplar icon is shown in [Figure 3-4](#).

Figure 3-4 *The Exemplar Icon*



7. To view the attributes for an exemplar, expand the exemplar and click its Attributes subobject. This displays the object definition's property pages in Plan Editor's detail view.
8. To view the grants associated with an exemplar, expand the exemplar and click its Grants subobject (note that the Grants subobject is not displayed for object types that do not participate in grants). This displays the object definition's Grants property page in Plan Editor's detail view. By default, when you include an exemplar in a plan, all the grants associated with the exemplar object are included in the plan, which means when the object definition is reproduced at a destination database, the object's grants are reproduced, if possible. A grant will be applied if the objects that reference the grant already exist or will be created at the destination database when the change plan's script is executed at the destination database.

Select one or more of an exemplar's grants and click the **Exclude** button to exclude those grants from the plan. If you decide later that you want to include

one or more excluded grants for an exemplar in the plan, select those grants, then click the **Include** button to include them in the plan.

After you include a specific grant for an exemplar in a plan, it is possible for the same grant to be modified in the database. In this case, the Refresh button becomes available when you select that grant on the Grants page. If you want to update the grant in the plan to match the grant in the database, select the grant, then click the **Refresh** button.

If no grants are associated with the exemplar, the Grants page does not display any grants.

9. To view the dependencies and dependents of an exemplar, expand the exemplar and click its Dependencies subobject. This displays the Dependencies and Dependents property pages for the exemplar.

The Dependencies page displays the objects that the exemplar depends on. Each dependency object definition on the Dependencies page should be added to the plan, except for those object definitions that already exist at the destination database. For example, suppose you add an exemplar for a trigger to a plan, and the trigger refers to a table that is not in the plan and which does not exist at the destination database. In this case, you should manually add an exemplar for the referenced table to the plan, otherwise the trigger will not be created at the destination database. To manually add a dependency object to the plan, select the object definition and click **Add to Plan**. Objects that are already in the plan are unavailable.

The Dependents page displays the objects that depend on the exemplar. You can use this page to locate other object definitions that are related to the exemplar, and, if you wish, manually add them to the plan. To manually add a dependent object definition to the plan, select the object definition and click **Add to Plan**. Objects that are already in the plan are unavailable.

By default, objects on the Dependencies and Dependents pages are displayed in a tree. If you click **Show List**, the objects on the selected page will be displayed in a list instead of a tree. Click **Show Tree** to display the objects in a tree again.

Click **Generate Report** to generate a report about the dependency and dependent objects. The report can be generated in HTML or comma-separated values format.

10. You can view and, if desired, change the value of one or more of the plan's propagation options. To do so, click on the plan name in the Plan Editor tree. In the detail view, click the Propagation Options tab to view and, if desired, modify one or more values for the plan's propagation options. The values of

the propagation options for a change plan determine how exemplars in the plan are applied when a script generated from the plan is run at a destination database.

If your plan contains table exemplars, you can reproduce both the table definitions for the exemplars and the data associated with the table definitions at a destination database. To do so, select the **Copy Table Data** option. If you want to reproduce only the definitions for the table exemplars, do not select the **Copy Table Data** option.

See the online help for a complete list of the propagation options and a description of how their values affect the application of a change plan's exemplars at a destination database.

Creating a Modified Exemplar with Plan Editor

You may want to edit the attributes of an exemplar definition before propagating it (instead of creating an exemplar definition to propagate, then manually editing the definition's attributes after it has been reproduced at a destination database). For example, you can modify the storage parameters for exemplar definitions before propagating the definitions from a production database to a test database. After the propagation operation completes, the propagated definitions will already have the desired storage parameters for the new destination. An exemplar with attributes that have been changed since the exemplar was created is referred to as a modified exemplar.

To modify an exemplar, thereby creating a modified exemplar:

1. In the Plan Editor tree, expand the **Change Requests** folder.
2. Expand the object type folder that contains the exemplar (for example, if the exemplar is a table definition, expand the **Tables** folder).
3. If the exemplar is a schema object, expand the folder for the schema.
4. Expand the exemplar.
5. Click the exemplar's Attributes subobject, which displays the exemplar's attributes in property pages in the detail view.
6. On the General page for the exemplar, choose **Use Modified Exemplar**, which enables you to make changes to the exemplar's attributes on the property pages.

Note: The following changes to an exemplar definition are not supported:

- Changing the name of the original exemplar definition
 - Changing the exemplar's schema
-
-

Both DB Propagate and Plan Editor allow you to create modified exemplars. However, Plan Editor offers more options for applying modified exemplars in change plan scripts at a destination database. See "[Generating a Script with Plan Editor](#)" on page 3-15 for more information.

Selecting a Destination Database with Plan Editor

To select the destination database where you want the plan to be executed:

1. In Plan Editor, choose **Plan->New Destination**.
2. On the General page of the Create Destination dialog box, select a destination database from the list of available destinations, and, optionally, supply a description for the database.
3. Click **Create**.

Understanding Script Generation

After a destination database has been selected, a script can be generated from the change plan. The script generated from the change plan (not the plan itself) will be run against the destination database. During the initial stage of script generation, the Oracle Change Management Pack application examines the structure and definitions in the destination database so that it can generate a script designed exclusively for execution against the destination database.

When you use a single plan to generate scripts for several databases that have different structures and definitions, a different script is generated for each database. This is because Plan Editor and the other change applications take each destination database's structure and definitions into account when generating the script.

For example, suppose your plan contains an exemplar for table Table_2, and you generate two scripts, one to run against destination database DB_1 and the other to run against destination database DB_2. If table Table_2 does not exist in database DB_1, the script generated for DB_1 will include statements that define table Table_2. If a different version of table Table_2 already exists in database DB_2, the script

generated for DB_2 will include statements to make the definition of table Table_2 in database DB_2 match the exemplar for table Table_2.

After Plan Editor generates a script for a destination database, you can view and, optionally, edit the script. An impact report is also created when the script is generated. You should view the impact report to determine the impact of executing the script at the destination database. The impact report provides a summary of the number and types of objects that will be modified when the script executes at the destination database. The impact report also shows warnings and errors, including a description of the requested operations that cannot be performed at the destination database, for example, a request to drop a column that no longer exists at the destination database. The other change applications also generate scripts that can be viewed and edited.

If you modify a plan's change requests after you have generated one or more scripts for the plan, the scripts that have been generated are considered to be obsolete scripts. When you try to execute an obsolete script, the Change Management Pack application you are using displays a message that advises you that the plan was modified after the script was generated and confirms whether you want to execute the script anyway. It is prudent to generate a new script from the modified plan instead of executing an obsolete script.

Generating a Script with Plan Editor

To generate a script for the destination database:

1. Expand the Destinations folder.
2. Expand the destination database.
3. Click the Script subobject. This displays the script property pages in the detail view, as shown in [Figure 3-5](#) on page 3-17.
4. Click the Options tab in the detail view. On the Options page, you can:
 - Map schemas in the source database to their corresponding schemas in the destination database. By default, change requests specified for database objects in a source schema are applied to a destination schema with the same name. You only need to map schemas when you want change requests for the object definitions in a schema in the source database to be applied to a destination schema with a different name. For example, if your plan includes change requests created for the SALES table in the FINANCE schema and you want to apply those changes to the SALES table in the FINANCE_V2 schema, then you need to map the FINANCE schema to the FINANCE_V2 schema. To map two schemas, select the source schema from

the source database list and select the destination schema from the destination database list, then click the Down arrow.

Note: Schema mapping is not applied to SQL or PL/SQL definitions of views, check constraints, triggers, and so on. For example, if a schema mapping XXX -> YYY exists, references to schema XXX in a view definition are not changed to YYY.

- Specify a scratch tablespace that Plan Editor can use for script operations that require temporary storage of data. For example, renaming a tablespace requires a scratch tablespace because all the data in the tablespace must be stored temporarily while the first tablespace is dropped and recreated. Other operations, such as operations on a table, require either enough storage space in the table's tablespace for two copies of the original table or a scratch tablespace to contain the table copy.
- Specify a commit interval to be used when the script runs. The specified value dictates the number of rows the script will process before performing a commit operation.

The Options page and its features are also available from other change applications besides Plan Editor.

5. Click the Advanced Options tab in the detail view. On the Advanced Options page, you can specify how the change plan's modified exemplars will be applied at the destination database when the script executes. The two choices are to apply either:
 - The original exemplars and the specified changes (this is the default)
 - Only the changes specified for the exemplars

If you choose to apply only the changes specified for the modified exemplars, there are several suboptions that allow you to select which changes to apply. See the online help for the Advanced Options page for more information about the options and suboptions for applying changes specified for modified exemplars.

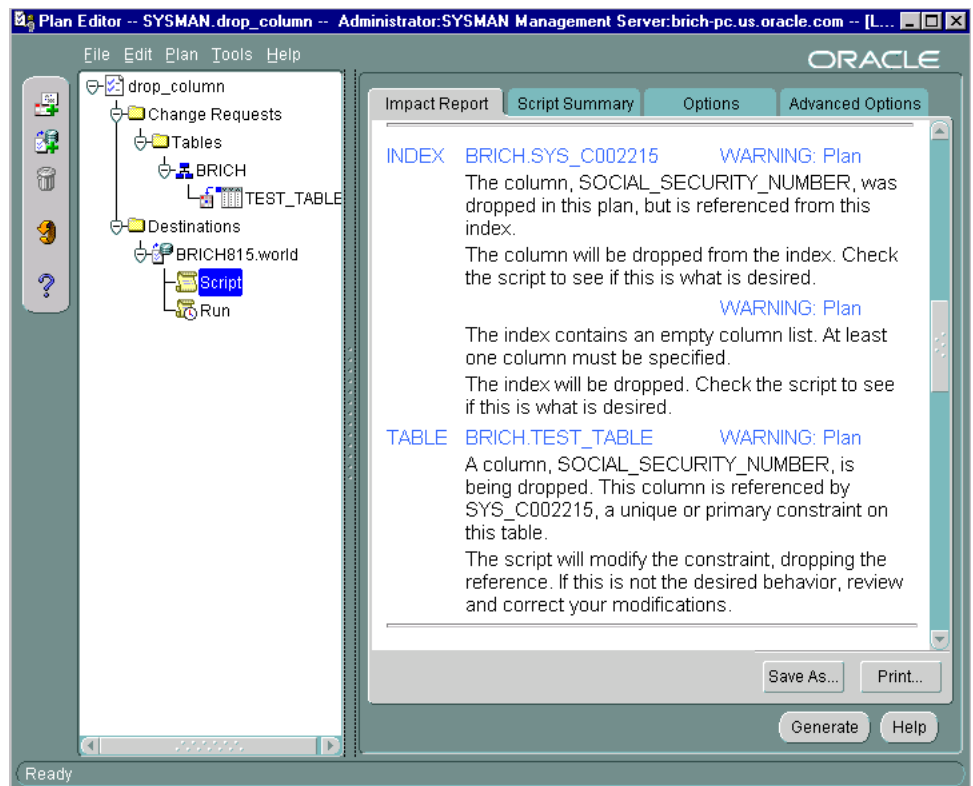
Modified exemplars can be created using Plan Editor or DB Propagate. However, the Advanced Options page is not available when you use DB Propagate to generate a script. Therefore, scripts generated by DB Propagate for plans with modified exemplars use the default method of applying modified exemplars (the script applies both the original exemplars and the specified changes at the destination database).

If you use DB Propagate to create a plan with modified exemplars and then generate a script for it, you might decide that you do not want the default option for applying modified exemplars. If so, edit the plan originally created in DB Propagate with Plan Editor, click the Script subobject, select the other option and the desired suboption on the Advanced Options page, then generate the script again and execute it using Plan Editor.

6. Click the **Generate** button to generate the script and the impact report that describes the impact of executing the script at the destination database. Work-in-progress messages display while the script is being generated.

Figure 3–5 shows a fully expanded change plan in Plan Editor with the Script subobject selected and the Impact Report page displayed.

Figure 3–5 A Fully Expanded Change Plan



Viewing the Impact Report and Script Summary with Plan Editor

Oracle Change Management Pack change applications create an impact report and a script summary when they generate a script.

1. Click the Impact Report tab to display the Impact Report page. [Figure 3-5](#) on page 3-17 shows an impact report displayed in Plan Editor.

View the impact report to determine the impact of executing the generated script at the destination database. The impact report provides a summary of the number and types of objects that will be modified when the script executes at the destination database. The impact report also shows warnings and errors, including a description of the requested operations that cannot be performed at the destination database, for example, a request to drop a column that no longer exists at the destination database.

Script errors must be fixed. Evaluate the errors, take corrective action, then regenerate the script.

Script warnings should be read and the appropriate action taken before you attempt to execute the script. Some warnings are informational, for example, a message that dropping a column will cause an index to be dropped, as shown in [Figure 3-5](#). To save the impact report to a file, click **Save As**. To print the impact report, click **Print**.

2. Click the Script Summary tab to display the Script Summary page.

View the script summary on the Script Summary page. The script summary contains the SQL statements and non-SQL operations that will be executed at the destination database to implement the plan's change requests. The actual script includes both the SQL statements from the script and OraTCL statements. To save the script summary to a file, click **Save As**. To print the script summary, click **Print**.

To edit the actual script (not the script summary), click the **Edit script** button on the Script Summary page. This displays the script in the Edit Oracle Tcl Script dialog box. Note that the actual script can be very difficult to understand, and editing it may result in undesired changes that you cannot undo after the script is executed. To save the script to a file, click the **Save As** button in the Edit Oracle Tcl Script dialog box.

A script that has been edited can be executed even if it has script generation errors.

Note: Scripts produced by Oracle Change Management Pack applications can only be run using Oracle Change Management Pack applications, the Oracle Change Management Pack command line interface, or the Oracle Enterprise Manager job system. See ["execute command"](#) on page A-10 for more information about executing a script using the command line interface. See the online help for more information about executing a script using the job system.

If the impact report or the script summary is unacceptable, you can take one or more of the following actions:

- Modify the plan by modifying the directives, exemplars, propagation options, (or all of these), then save the plan, and generate a new script.
- Modify the destination database (add, modify, or delete object definitions, for example), then generate a new script.
- Change the schema mappings on the Options page, then generate a new script.
- Edit the script, then execute the script. To edit a script, click **Edit script** on the Script page. Note that the actual script can be very difficult to understand, and editing it may result in undesired changes that you cannot undo after the script is executed.

Understanding Script Execution

To ensure that you can undo the execution of a script, Oracle Change Management Pack change applications make a copy of the old data when it is needed for recovery purposes. The recovery data is stored in temporary tables that look like the original tables but have different names.

Oracle Change Management Pack change applications give you the option of keeping or undoing the changes made when a script is executed.

If you keep the changes, the temporary tables used by the recovery script are deleted, making the changes permanent.

If you undo the changes, the recovery script uses the recovery data to return the user tables and data to their original state.

Executing the Script with Plan Editor

To execute the script against the destination database:

1. Expand the destination database and select the Run subobject in the Plan Editor tree.
2. On the Execution Log page in the detail view, click **Execute** to run the script immediately.

Note: To execute a script generated by any of the Oracle Change Management Pack applications, use an account for which DBA privileges are set as the preferred credentials.

3. You can examine the execution log on the Execution Log page during or after script execution. The execution log displays messages, including the status of the script execution ("Script execution succeeded" or "Script execution failed"). To save the execution log to a file, click the **Save As** button.
4. After executing a script, the Execution Log page lets you keep or undo the changes made by the script to object definitions at the destination database:
 - When you click **Keep**, Plan Editor deletes temporary tables used by the recovery script, making the changes at the destination database permanent.
 - When you click **Undo**, Plan Editor uses the recovery script to undo the changes made at the destination database.

Note: There are a small number of attribute changes for which undo operations are not present in the recovery script. The common characteristics of these cases is that the original change can be done in a single ALTER statement and the undo operation requires multiple steps. In all cases where an undo operation is not included in the recovery script:

- You are clearly warned, both in the impact report and if you execute the recovery script.
- You can still use a change application to perform the undo operation, but you must do this as a separate step.

Suppose, for example, that a change plan makes several changes, including adding a column to a table. The impact report warns you that the recovery script will not drop the added column. (Note that the other changes can be undone.) If you execute the recovery script, you are warned again that the added column was not dropped. You can then use another change application (such as DB Quick Change) to drop the column as a separate step.

The attribute changes for which undo operations are not present in the recovery script are:

- adding a column to a table
 - modifying a column's datatype to specify a larger size for the column
 - adding a datafile to a tablespace
 - specifying a date for the Start Date or Next Date fields for a snapshot
-
-

If you want to execute the plan's change requests against a different destination database, select a new destination database, then generate, view, and execute a new script against the database.

You can also execute a script using:

- The Oracle Enterprise Manager job system. See the online help for more information.
- The Oracle Change Management Pack command line interface. See "[execute command](#)" on page A-10 for more information.

All the change applications allow you to execute scripts generated by that application. Plan Editor can execute scripts generated by any change application.

Dealing with Script Execution Errors

Two common causes of script execution errors are:

1. Stale scripts

When the Oracle Change Management Pack translator generates a script for a destination database, it takes the current structure of the database into account. If objects at the destination database are deleted or modified after the script is generated, the script is considered to be a stale script. Errors can occur during the execution of a stale script or recovery script. For example, if a particular user is removed from a destination database before you run a script, the execution log may display an error message such as this after the statement that generated the error message:

```
ORA-01918: user 'GEORGE' does not exist
```

2. Problems that the Oracle Change Management Pack translator does not anticipate when it generates a script, for example, insufficient space in a tablespace to carry out the requested changes.

If you suspect that a script execution error is caused by a problem with the Oracle Change Management Pack translator, please contact Oracle Worldwide Customer Support.

To have Oracle Change Management Pack predict script execution failures (which is the default behavior), the `OCM_FAILURE_PREDICTION` property in the `ORACLE_HOME/sysman/config/omsconfig.properties` file must be enabled (where `ORACLE_HOME` is the directory in which Oracle Enterprise Manager is installed). The property is enabled when its value is set to "true" (case insensitive) or when the property does not exist in the `omsconfig.properties` file. When failure prediction is enabled, Oracle Change Management Pack change applications perform resource checking (for example, they check for sufficient space and quota to make copies of tables or to move items from one tablespace to another) during script generation. Resource warnings are reported in the impact report. Script generation takes longer when failure prediction is enabled.

If the `OCM_FAILURE_PREDICTION` property is present in the `omsconfig.properties` file and has a value other than "true," then resource checking and script failure prediction does not occur.

When error messages occur during script execution, the best ways to fix the problem are:

- Click the **Undo** button to undo the changes. Then, regenerate the script.
- Correct the error at the destination database (for example, by creating a user again or by increasing the size of a tablespace), then click **Continue** to continue executing the script. If you specify a scratch tablespace, you must generate a new script and execute the new script.
- Edit the script and click **Continue** to continue executing the script. This is the least preferable option, because editing the script may result in undesired changes that are not reversed when you press **Undo**.

When error messages occur during recovery script execution, the best ways to fix the problem (in order of preference) are:

1. Correct the error at the destination database (for example, by creating the user again), then click **Undo** to continue executing the recovery script.
2. Edit the recovery script, then click **Undo** to continue executing the recovery script.

Command Line Interface Appendix

Oracle Change Management Pack provides a command line interface that is available on the Windows NT and UNIX operating systems. This appendix provides the following sections that provide more information about the command line interface:

- ["Possible Uses for the Command Line Interface"](#) on page A-1
- ["Understanding the Notation Used for the Command Syntax"](#) on page A-2
- ["Requirements for Using the Command Line Interface"](#) on page A-2
- ["Scheduling the Execution of Command Line Interface Scripts"](#) on page A-4
- ["Using the Command Line Interface Commands"](#) on page A-4
- ["Status Codes Returned by the Command Line Interface"](#) on page A-14

Note: Before using the command line interface, you should be familiar with the Oracle Change Management Pack concepts and applications described in the previous chapters of this manual.

Possible Uses for the Command Line Interface

You can use the Oracle Change Management Pack command line interface to perform various Change Management Pack tasks without accessing the Oracle Change Management Pack applications. Some of the tasks that you can perform using the command line interface include:

- Capturing the definitions of a set of database objects each night, using the same baseline specification

- Comparing the definitions of a set of database objects to those of a different schema or database, or a baseline, each night
- Generating scripts during off hours by applying a single plan to each of a set of destination databases, and then executing the script at each destination if the generation completes without error

Understanding the Notation Used for the Command Syntax

The reference sections that describe each command in the Oracle Change Management Pack's command line interface use the following notation to describe command syntax:

Square brackets [...] denote optional elements.

Braces {...} denote a required element which can be one of two or more options separated by vertical bars |.

Square brackets enclosing options separated by vertical bars indicate an optional element which can be one of two or more options.

Italics denote elements to be substituted by the user, or elements that are further defined in the command description section.

All other characters are keywords and punctuation that should be typed as shown in the command descriptions.

Requirements for Using the Command Line Interface

Important: Before issuing command line interface commands using the Solaris operating system, make sure that your default directory is the ORACLE_HOME/bin directory (where ORACLE_HOME is the directory in which Oracle Enterprise Manager is installed).

The command line interface on Solaris only recognizes command line interface commands that are issued in the ORACLE_HOME/bin directory.

When you enter the command line interface commands interactively at the MS-DOS prompt or UNIX prompt, preface each command with the ocm keyword, for example:

ocm *command-name*

When you use the command line interface, replace *command-name* with the name of a command line interface command.

When you use the command line interface, do not leave any blank spaces between a command line option and the argument specified for the option. For example, in the following command, the `-d`, `-s`, `-e`, and `-id` options are followed immediately by their respective arguments, `forward`, `cleanup`, `undo`, and `fred`:

```
ocm execute plan15 corporate.world -dforward -scleanup -eundo -idfred
```

Note: There is no continuation character for command line interface commands. Therefore, when you enter a command line interface command interactively at the command prompt, enter the entire command before pressing Return.

Similarly, when entering command line interface commands in a script file, do not enter a carriage return for a command until you have entered the complete command.

Understanding Case-Sensitivity Requirements for the Command Line Interface

The command line interface's case-sensitivity requirements are:

- Commands, options, and keywords must be specified in lowercase.
- The case of a change plan, baseline, or comparison name specified with the command line interface must exactly match the case of that object's name as it appears in the Change Manager application.
- Database names are not case-sensitive.
- The string that you specify for the `-id` option's *identity* argument is case-sensitive.

Using the Command Line Interface with the Oracle Management Server

You can use the command line interface with or without the Oracle Management Server (OMS). When you execute the login command, a login dialog box is displayed that allows you to specify whether or not to use the Oracle Management Server. See "[login command](#)" on page A-5 for more information on the login command.

Scheduling the Execution of Command Line Interface Scripts

With the NT operating system, you can enter the command line interface commands in a .BAT file. You can then parameterize the .BAT file and run it at specified times and/or intervals using operating system facilities. The commands return a status code, which allows flow-of-control within the containing .BAT file. If you are entering the commands in a .BAT file, use the MS-DOS call command to preface calls to the command line interface commands, for example:

```
call ocm command-name
```

Similarly, with the UNIX operating system, you can enter the command line interface commands in a script. You can parameterize the script file and run it at specified times and/or intervals using operating system facilities. The commands return a status code, which allows flow-of-control within the containing script file.

Using the Command Line Interface Commands

[Table A-1](#) lists each command line interface command and provides a brief description of the command's function:

Table A-1 Overview of Command Line Interface Commands

Command	Use of the Command	For More Information
login	Establishes the credentials that an administrator will use during a command line interface session.	See " login command " on page A-5
logout	De-establishes the administrator credentials for the specified administrator.	See " logout command " on page A-7
capture	Captures a new baseline for an existing baseline specification.	See " capture command " on page A-7
compare	Compares databases, baselines, or a baseline and a database using an existing comparison specification.	See " compare command " on page A-8
generate	Generates a script from a change plan.	See " generate command " on page A-9

Table A-1 Overview of Command Line Interface Commands (Cont.)

Command	Use of the Command	For More Information
execute	Executes a script, undoes changes made by a previously-executed script, or cleans up after the successful execution of a script.	See " execute command " on page A-10

login command

The login command is used to establish or change administrator credentials for a command line interface session. You must establish administrator credentials for a command line interface session, otherwise the command line interface commands that you issue will fail.

The syntax for the command is:

```
ocm login -ididentity
```

The string that you choose for the *identity* argument is used to uniquely identify you during a command line interface session. The identity string is case-sensitive.

For example, you can specify a login command like this one:

```
ocm login -idfred
```

When the login command is executed, a login dialog box displays that prompts you whether you want to use the command line interface logged into the Oracle Management Server, or whether you want to use the command line interface standalone (not logged into the Oracle Management Server). After you choose the mode in which you want to run the command line interface, the dialog box prompts you for the necessary credentials for that option:

- If you choose to run the command line interface logged into the Oracle Management Server, you must specify your Oracle Enterprise Manager administrator's name, your administrator password, and the name of the node on which the Oracle Management Server is running.
- If you choose to run the command line interface standalone (without logging into the Oracle Management Server), you must specify the Username, Password, and Service (database) credentials of the user that owns the schema used for your standalone repository.

Important: Be aware of the following considerations when you use the login command:

- Before you run a script that contains command line interface commands, you must first issue the login command outside of the script to establish the administrator credentials that are referenced inside the script. For example, if a script includes commands that use the `-idfred` argument, the `ocm login -idfred` command must be issued prior to the script being invoked so that credentials for the administrator identified by `-idfred` can be established before the script is run.
 - To execute the login command on a remote machine, your local machine's display must be appropriately set to allow the login dialog box to display on your local machine. Then you can use the login dialog box to establish the administrator credentials to be used for the command line interface session.
-
-

The command line interface associates the string that you specified for the *identity* argument (fred in the example in this section) with:

- Your decision to run the command line interface connected to the Oracle Management Server or standalone.
- The credentials you specified for the option you selected. When you execute other command line interface commands, specify the same identity string.

Because of this association, when you execute other commands that include the same *identity* argument, the command line interface knows:

- The administrator credentials that are specified for that identity.
- Where (in what repository) to store the Change Management Pack objects created during the command line session for that administrator.

The login command sets the credentials in the administrator's environment that are used until one of the following occurs:

- The operating system session is terminated.
- The `ocm logout` command is used to de-establish the administrator credentials for a specific administrator.

logout command

The logout command de-establishes the administrator credentials established with an earlier login command.

The syntax for the command is:

```
ocm logout -iidentity
```

For example, the following command ends the command line interface session for the administrator identified as fred:

```
ocm logout -idfired
```

See "[login command](#)" on page A-5 for more information about the *identity* argument.

capture command

The capture command captures a new baseline for an existing baseline specification.

The syntax for this command is:

```
ocm capture [-l] baseline-specification-name -iidentity
```

The *baseline-specification-name* must be the name of an existing baseline specification. The *baseline-specification-name* must exactly match the case of the baseline specification's name as it appears in the Change Manager application. Enclose the *baseline-specification-name* within double quotation marks if it contains special characters such as backslashes, asterisks, or spaces, for example:

```
ocm capture "Baseline Specification Name With Spaces" -idfired
```

The command in this example uses the baseline specification named "Baseline Specification Name With Spaces" to capture a new version of that baseline. The baseline is stored in the repository for the administrator identified as fred. Note that the identity specified with the *-id* argument was established previously with the login command. See "[login command](#)" on page A-5 for more information about the *identity* argument.

The optional *-l* option is used to enable logging for the capture command. When logging is enabled, the same messages that would be displayed in Create Baseline during a baseline creation operation are displayed in the DOS window or UNIX window while the capture command is executing. If the *-l* option is not specified on the command line, no messages are displayed in the DOS window or UNIX window while the capture command is executing.

Baselines cannot be viewed using the command line interface. To view a baseline Change Manager application, select the baseline in the navigator tree, and choose **Object->View Baseline**.

In some cases, the capture command may return a status code other than success. For example, the output of the command may be a non- success status code if the baseline specification referred to in the capture command does not exist. See [Table A-2](#) on page A-15 for more information on exit status codes that can be returned by the command line interface.

compare command

The compare command compares databases and/or baselines using an existing comparison specification.

The syntax for this command is:

```
ocm compare [-l] comparison-specification-name -ididentity
```

Each comparison specification has a series of comparisons associated with it; the compare command creates another comparison in the series. The *comparison-specification-name* must be the name of an existing comparison specification. The *comparison-specification-name* must exactly match the case of the comparison specification's name as it appears in the Change Manager application. Enclose the *comparison-specification-name* within double quotation marks if it contains special characters such as backslashes, asterisks, or spaces, for example:

```
ocm compare "Comparison Specification Name With Spaces" -idfred
```

The command in this example uses the comparison specification named "Comparison Specification Name With Spaces" to capture a new version of that comparison. The comparison is stored in the repository for the administrator identified as fred. Note that the identity specified with the `-id` argument was established previously with the `login` command. See "[login command](#)" on page A-5 for more information about the *identity* argument.

The optional `-l` option is used to enable logging for the compare command. When logging is enabled, the same messages that would be displayed in the graphical user interface during a comparison creation operation are displayed in the DOS window or UNIX window while the compare command is executing. If the `-l` option is not specified on the command line, no messages are displayed in the DOS window or UNIX window while the compare command is executing.

Comparisons cannot be viewed using the command line interface. To view a comparison created using the compare command, select the comparison in Change Manager and choose **Object->View Comparison**.

In some cases, the compare command may return a status code other than success. For example, the output of the command may be a non- success status code if the comparison specification referred to in the compare command does not exist. See [Table A-2](#) on page A-15 for more information on exit status codes that can be returned by the command line interface.

generate command

The generate command generates a script, script summary, and impact report. Generation is the process of applying the change requests in a plan to a destination database, producing a script that will carry out the change requests, a script summary, and an impact report that describes the changes that will be made when the script is executed at the destination database.

The syntax for this command is:

```
ocm generate [-l] plan destination-database -ididentity
```

The *plan* keyword identifies the plan for which you want to generate a script, script summary, and impact report. The *destination-database* keyword identifies the database at which the generated script will be executed (the generate command does not execute the script; the script can be executed later at the destination database using the execute command).

The plan that you specify for the *plan* keyword must already exist, and the destination database that you specify for the *destination-database* keyword must already be specified in that plan. The *plan* name must exactly match the case of the plan's name as it appears in the Change Manager application. The *destination-database* name is not case-sensitive, but it must match the name of the destination database as it appears in the Change Manager application.

Enclose the plan name within double quotation marks if it contains special characters such as backslashes, asterisks, or spaces, for example:

```
ocm generate "Plan Name With Spaces" personnel_db -idfred
```

The command in this example uses the generates a script, script summary, and impact report from the plan named "Plan Name With Spaces" for the personnel_db database. The script, script summary, and impact report are stored in the repository for the administrator identified as fred. Note that the identity specified with the -id

argument was established previously with the login command. See "[login command](#)" on page A-5 for more information about the *identity* argument.

The optional `-l` option is used to enable logging for the generate command. When logging is enabled, the same messages that would be displayed in the graphical user interface during a script generation operation are displayed in the DOS window or UNIX window while the generate command is executing. If the `-l` option is not specified on the command line, no messages are displayed in the DOS window or UNIX window while the generate command is executing.

Scripts, script summaries, and impact reports cannot be viewed using the command line interface. To view the script, script summary and impact report generated with the generate command, start the Change Manager application, select the plan, and choose **Object->Edit Change Plan**. In Plan Editor, expand the tree and the destination database folder, then click the Script subobject. In the detail view, click the Script Summary tab to view the Script Summary page, then click **Edit Script** to view or edit the script itself. In the detail view, click the Impact Report tab to view the Impact Report page. In Plan Editor's detail view, you can also click the Options tab, then use the Options page to specify a commit interval, schema mapping, and/or a scratch tablespace to use when the script is executed at the destination database. Similarly, you can click the Advanced Options tab, then use the Advanced Options page to specify how modified exemplars should be applied when the script is executed at the destination database.

In some cases, the generate command may return a status code other than success. For example, the output of the command may be a non- success status code if the impact report contains warnings or errors, or if the script cannot be generated. See [Table A-2](#) on page A-15 for more information on exit status codes that can be returned by the command line interface.

execute command

The execute command executes a script, undoes a previously-executed script, or cleans up after the successful execution of a script. The script must be in an appropriate state for the requested operation or the command will fail.

The syntax for this command is:

```
ocm execute [-l] plan destination-database
           [ [ -d{forward|backward|cleanup} ]
           [ -s{cleanup|exit} ]
           [ -e{undo|exit} ] ]
           -identity
```

The *plan* keyword identifies the name of the plan for which the script should be executed. The *destination-database* keyword identifies the database at which you want the script to execute. A script must already be generated for this plan and this destination database, otherwise the execute command will fail. The *plan* name must exactly match the case of the plan's name as it appears in the Change Manager application. The *destination-database* name is not case-sensitive, but it must match the name of the destination database as it appears in the Change Manager application. If the plan name contains special characters such as backslashes, asterisks, or spaces, enclose it within double quotation marks.

The optional `-l` option is used to enable logging for the execute command. When logging is enabled, the same messages that would be displayed in the graphical user interface during a script execution operation are displayed in the DOS window or UNIX window while the execute command is executing. If the `-l` option is not specified on the command line, no messages are displayed in the DOS window or UNIX window while the execute command is executing.

Note that the identity specified with the `-id` argument is an identity established previously with the login command. See "[login command](#)" on page A-5 for more information about the *identity* argument.

Execution logs cannot be viewed using the command line interface. To view the execution log for a script whose changes have been kept or undone with the execute command, start Change Manager, select the plan, and choose **Object->Edit Change Plan**. In Plan Editor, expand the tree and the destination database folder, then click the Run subobject. In the detail view, view the execution log on the Execution Log page.

Specifying the Direction in Which to Execute the Script

Use the `-d` option to specify the direction in which the script should be executed or to specify that you want to clean up after the successful execution of the script. The `-d` options are:

- forward

The forward option executes the script.

- backward

The backward option executes the recovery script that undoes the changes made by the script. Use this option if the script failed or you do not like the changes made by the script. Using this option is equivalent to selecting the Undo option in Plan Editor or one of the other change applications after executing a script.

- **cleanup**

The cleanup option keeps the changes made by the script and removes the temporary data that the recovery script uses to undo changes made by the script. Use this option only after the script has successfully executed in the forward direction. Using this option is equivalent to selecting the Keep option in Plan Editor or one of the other change applications after executing a script. After using this option, you cannot use the backward option.

By default, the script is executed in the forward direction (from the start or from the last successfully completed step, as applicable).

Specifying Completion Actions for a Script Execution

The -s and -e options are used to specify completion actions.

Use the -s option to specify the action to be taken if the script executes successfully. The two -s options are:

- **cleanup**

When the cleanup option is specified with the -s option and the script executes successfully, the changes made by the script are kept and the temporary data that the recovery script uses to undo changes made by the script is removed.

The cleanup option only makes sense when the script is executed in the forward direction.

- **exit**

When the exit option is specified with the -s option and the script executes successfully, the changes made by the script execution are not kept or undone, which means your script execution results in a pending task. To complete this pending task, you need to specify later whether you want to keep or undo the changes. You can use one of the change applications or issue another execute command to specify whether to keep or undo the changes. See "[Viewing and Completing Pending Tasks](#)" on page 2-12 for more information on pending tasks and how to complete them.

If you do not use the -s option, the default action is to exit on success.

Use the -e option to specify the action to be taken if errors occur when the script executes. The two -e options are:

- **undo**

When the undo option is specified with the -e option and errors occur when the script executes, the recovery script executes, undoing the changes made by the

script. Using this option is equivalent to selecting the Undo option in Plan Editor or one of the other change applications after executing a script.

The undo option only makes sense when the script is executed in the forward direction.

- exit

When the exit option is specified with the -e option and errors occur when the script executes, the changes made by the script execution are not kept or undone, which means your script execution results in a pending task. To complete this pending task, you need to specify later whether you want to keep or undo the changes.

When an error occurs during script execution, you can use several different approaches for determining the cause of the error, including:

- Use [Table A-2](#) on page A-15 to look up the meaning of the exit status code returned for the execute command.
- If you used the -l option with the execute command, look at the logging output that was generated by the execute command.
- Use Plan Editor or one of the change applications to view the execution log for the script. See "[Executing the Script with Plan Editor](#)" on page 3-20 for more information on viewing an execution log.

After determining the cause of the execution error, you can either keep or undo the changes to complete the pending task, or you can make changes to the script or destination database that will allow you to execute the script successfully. See "[Dealing with Script Execution Errors](#)" on page 3-22 for more information on dealing with script execution errors.

If you do not use the -e option, the default action is to exit on error.

Examples of Using the Execute Command

to execute a script in the forward direction, keeping the changes if the execution completes successfully and undoing the changes if the execution fails, you would use a command like the following:

```
ocm execute plan15 corporate.world -dforward -scleanup -eundo -idfred
```

To execute a script in the forward direction, undoing the changes if there is an error and exiting on success, you would use a command like the following:

```
ocm execute plan15 corporate.world -eundo -idfred
```

The following command executes a script in the forward direction, undoing the changes if there is an error and exiting on success. It also enables logging in the DOS or UNIX window while the execute command is executing:

```
ocm execute -l plan15 corporate.world -eundo -idfred
```

To keep the changes made by a script that has already executed successfully in the forward direction and exit on success or error, use a command like the following:

```
ocm execute plan15 corporate.world -dcleanup -idfred
```

You can also execute a script in the backward direction, which undoes the changes made a script that has been executed in the forward direction. The backward option works only if the script was previously executed in the forward direction, either to completion or to a point of failure. To do this, exiting on either success or error, use a command like the following:

```
ocm execute update_plan my_database.world -dbackward -idfred
```

Enclose the plan name within double quotation marks if it contains special characters such as backslashes, asterisks, or spaces, for example:

```
ocm execute "Plan Name With Spaces" personnel_db -dforward -scleanup -eundo -idfred
```

In some cases, the execute command may return a status code other than success. For example, the output of the command may be a non- success status code if the script for a plan and database is already executing when the execute command is issued.

The execute command also fails if the script is in an inappropriate state for the execute command that you specify. For example, if you issue the execute command with the -dcleanup option for a plan, then issue the execute command with the -dbackup option for that plan, the second execute command fails. See [Table A-2](#) on page A-15 for more information on exit status codes that can be returned by the command line interface.

Status Codes Returned by the Command Line Interface

After you execute each command line interface command, the command line interface returns an exit status code for the command that you entered.

[Table A-2](#) shows the meaning of each exit status value that can be returned.

Table A–2 Exit Status Codes Returned by the Command Line Interface

Exit Status	Description
0	Operation completed successfully.
1	Syntax error in command line.
2	The Oracle Management Server specified in the login command does not exist or the Oracle Management Server is not running.
3	There was an error when trying to login to the Oracle Management Server. Make sure that you specify correct credentials for the Oracle Management Server in the login command.
4	General, catchall error.
5	The baseline specified does not exist.
6	The comparison specified does not exist.
7	The change plan specified does not exist.
8	The script for that plan and database is already executing.
9	An error occurred during execution of the script.
10	Generation completed. Impact log contains warnings.
11	Generation completed. Impact log contains errors.
12	Generation failed. No script generated.
13	A problem occurred with the target database. For example, it does not exist.
14	Preferred credentials are not defined for the target database. Use the Oracle Enterprise Manager console to set preferred credentials for the database, then retry the operation.
15	You must use the login command to specify credentials for the Oracle Management Server before you can issue other command line interface commands.

To obtain more information about an error or warning status code returned after a generate or execute command, use the Change Management Pack graphical user interface to view the impact log or execution log.

Troubleshooting Appendix

This appendix describes different ways of troubleshooting problems that may occur when you are using Oracle Change Management Pack applications.

You may decide to contact an Oracle Worldwide Customer Support representative regarding a problem with Oracle Change Management Pack. If so, you may be asked to use one of the approaches described in this appendix to obtain information that will help the representative determine the source of the problem.

Using Environment Variables When Running Change Manager from the Command Line

You can set environment variables in the MS-DOS or UNIX command line before you run the Change Manager application from the command line. The following environment variables may provide information that is useful for troubleshooting and debugging Oracle Change Management Pack problems:

- `set ORACLE_OEM_JAVAMX=-mx<number>m`

The default is `-mx128m`. This environment variable specifies the maximum amount of virtual memory to be used by the Java Virtual Machine (in megabytes). This memory is not used unless needed. If the amount is exceeded, execution will be terminated. This environment variable can be used to increase the amount of memory for large script generations and other purposes.

- `set ORACLE_OEM_CLIENTTRACE=true`

This environment variable, which is for Windows NT only, displays troubleshooting information in an MS-DOS window. Before you set this environment variable, the `OCM_TRACE.ENABLED` property in the `ORACLE_HOME/sysman/config/ClientConfig.properties` file (described in [Table B-1](#) on page B-2) must be set to true, otherwise troubleshooting information is not

displayed in the MS-DOS window. Note that ORACLE_HOME is the directory in which Oracle Enterprise Manager is installed.

- set ORACLE_OEM_JAVARUNTIME=<directory where bin/java.exe is located>

You can specify this environment variable if you have a Java development environment. This environment variable can be useful when the system is hanging and you want to get a thread dump.

Enabling Tracing and Debugging for Oracle Change Management Pack

You can turn on tracing and debugging facilities for the Oracle Change Management Pack applications to help troubleshoot problems.

You can enter different properties in the ClientConfig.properties file in the ORACLE_HOME/sysman/config directory for Oracle Enterprise Manager to obtain more tracing and debugging information about Oracle Change Management Pack applications. [Table B-1](#) displays the properties and a description of each property.

By default, the trace files for Oracle Change Management Pack applications are written to the ORACLE_HOME/sysman/log directory for Oracle Enterprise Manager.

Note: The OCM_TRACE.ENABLED property in the ClientConfig.properties file must be set to true for the other properties whose names begin with OCM_TRACE in the ClientConfig.properties file to be meaningful.

Table B-1 *Using ClientConfig.properties to Enable Tracing and Debugging*

Property and Settings	Description
OCM_TRACE.ENABLED={true false}	If true, enables output of informational and problem trace messages.
OCM_TRACE_FILE.ENABLED={true false}	If true, enables output of trace information to a file. The setting of the OCM_TRACE_VERSIONING.ENABLED property determines the name of the trace file.

Table B-1 Using ClientConfig.properties to Enable Tracing and Debugging (Cont.)

Property and Settings	Description
OCM_TRACE_VERSIONING.ENABLED={true false}	<p>The OCM_TRACE_FILE.ENABLED property must be true for the OCM_TRACE_VERSIONING.ENABLED property to be meaningful.</p> <p>If the OCM_TRACE_VERSIONING.ENABLED property is true, a new version of the trace file is created for each run of the application. The file has a name in the format Oracle Change Manager_nnnnnn.log, where nnnnnn represents a series of numbers.</p> <p>If an Oracle Change Management Pack application is running in more than one session on your machine, examine the timestamp of each trace file to determine the trace file that is associated with each session. The trace file with the earliest timestamp is the trace file for the session that started first, and so on.</p> <p>If the OCM_TRACE_VERSIONING.ENABLED property is false or not used, the trace file has the name Oracle Change Manager.log, and the file is overwritten for each run of the application.</p>

Table B-1 Using ClientConfig.properties to Enable Tracing and Debugging (Cont.)

Property and Settings	Description
OCM_TRACE_DEBUG.ENABLED={true false}	If the OCM_TRACE_DEBUG.ENABLED property is true, general debugging information (such as property values) is output to the trace file while Change Manager runs.
OCM_TRACE_DEBUG_VERBOSE.ENABLED={true false}	If the OCM_TRACE_DEBUG_VERBOSE.ENABLED property is true, more detailed debugging information, including class names and line numbers, is output to the trace file while Change Manager runs.

Table B-1 Using ClientConfig.properties to Enable Tracing and Debugging (Cont.)

Property and Settings	Description
VDB_DEBUG={true false}	<p data-bbox="1032 296 1315 401">If the VDB_DEBUG property is true, database calls are output to the trace file.</p> <p data-bbox="1032 418 1315 604">This property is not specific to Change Management Pack applications. It can be used with other Oracle Enterprise Manager applications.</p> <p data-bbox="1032 621 1315 777">Use this property in the ClientConfig.properties file to capture database calls in the trace file when you are running Change Manager standalone.</p> <p data-bbox="1032 795 1315 1055">Use this property in the omsconfig.properties file in the \$ORACLE_HOME/sysman/config directory to capture database calls in the trace file when you are running Change Manager connected to an Oracle Management Server.</p> <p data-bbox="1032 1072 1315 1340">Use this property in both the ClientConfig.properties file and the omsconfig.properties file if you want to capture database calls in the trace file regardless of how you are running Change Manager.</p>

Table B-1 Using ClientConfig.properties to Enable Tracing and Debugging (Cont.)

Property and Settings	Description
VDB_VERBOSE_DEBUG={true false}	<p>If the VDB_VERBOSE_DEBUG property is true, more detailed information about database calls, including SQL statement issued and return values, is output to the trace file.</p> <p>This property is not specific to Change Management Pack applications. It can be used with other Oracle Enterprise Manager applications.</p> <p>Use this property in the ClientConfig.properties file to capture database calls in the trace file when you are running Change Manager standalone.</p> <p>Use this property in the omsconfig.properties file in the \$ORACLE_HOME/sysman/config directory to capture database calls in the trace file when you are running Change Manager connected to an Oracle Management Server.</p> <p>Use this property in both the ClientConfig.properties file and the omsconfig.properties file if you want to capture database calls in the trace file regardless of how you are running Change Manager.</p>

Index

A

- attribute changes not undone by a recovery script, 3-21
- attributes
 - modifying for a directive, 3-7
 - modifying for an exemplar, 3-13
 - viewing for a directive, 3-7
 - viewing for an exemplar, 3-11

B

- baseline
 - capturing definitions using Create Baseline, 1-9
 - capturing definitions using the command line interface, A-7
 - creating, 2-7
 - creating a new version, 2-8
 - creating using the command line interface, A-7
 - definition, 1-5
 - deleting, 2-7
 - deleting a previous version of, 2-8
 - exporting, 2-8, 2-14
 - importing, 2-8, 2-14
 - viewing, 1-9, 2-7
 - viewing a previous version of, 2-8
 - viewing history entries for, 2-8
 - working with a particular version of, 2-13
- Baseline Viewer
 - function of, 1-9

C

- capture command

- enabling logging for, A-7
- change management applications
 - starting from Change Manager, 2-6
- change management objects, 1-4
 - copying from one repository to another, 2-14
- Change Manager application
 - right mouse button support for, 2-6
 - running from the command line, 2-3
 - starting, 2-3
 - starting other applications from, 2-6
- change plan
 - creating, 2-10, 3-3
 - creating a new version, 2-10
 - defining change requests for, 3-4, 3-6
 - definition, 1-5
 - deleting, 2-10
 - deleting a previous version, 2-10
 - editing and modifying, 2-10
 - excluding grants from, 3-11
 - exporting, 2-10, 2-14
 - generating a script from, 3-15
 - generating a script using the command line interface, A-9
 - importing, 2-10, 2-14
 - including grants in, 3-12
 - naming, 3-4
 - renaming, 2-10
 - selecting a destination database for, 3-14
 - selecting the source database for, 3-4
 - viewing a previous version, 2-10
 - viewing and modifying the propagation options for, 3-12
 - viewing history entries for, 2-11
 - working with a particular version of, 2-13

- change request
 - defining a directive, 3-7
 - defining a modified exemplar, 3-13
 - defining a scoped directive, 3-8
 - defining an exemplar, 3-10
 - types, 3-4
- changes
 - for which undo operations are not present in the recovery script, 3-21
 - making, 1-7
 - tracking, 1-6
- ClientConfig.properties file
 - description of properties, B-2
 - location of, B-2
 - using to enable tracing and debugging for applications, B-2
- command line
 - running Change Manager from, 2-3
- command line interface
 - capture command, A-7
 - enabling logging for, A-7
 - compare command, A-8
 - enabling logging for, A-8
 - creating a baseline using, A-7
 - creating a comparison using, A-8
 - de-establishing administrator credentials for, A-7
 - entering commands in a .BAT file, A-4
 - establishing administrator credentials for, A-5
 - execute command, A-10
 - enabling logging for, A-11
 - examples of, A-13
 - executing a script using, A-10
 - exit status codes returned by, A-14
 - generate command, A-9
 - enabling logging for, A-10
 - generating a script using, A-9
 - login command, A-5
 - logout command, A-7
 - overview of commands, A-4
 - requirements for using, A-2
 - scheduling the execution of scripts, A-4
 - specifying completion actions for a script execution, A-12
 - specifying the direction in which to execute a script, A-11
 - understanding the notation for command syntax, A-2
 - uses for, A-1
 - using on UNIX, A-1
 - using on Windows NT, A-1
- compare command
 - enabling logging for, A-8
- Compare Database Objects application
 - function of, 1-7
 - overview of, 1-10
- comparison
 - creating, 1-10, 2-8
 - creating a new version of, 2-9
 - creating using the command line interface, A-8
 - definition, 1-5
 - deleting, 2-9
 - deleting a previous version of, 2-9
 - exporting, 2-9, 2-14
 - importing, 2-9, 2-14
 - naming, 1-10
 - saving, 1-10
 - viewing, 1-10, 2-9
 - viewing a previous version of, 2-9
 - viewing history entries for, 2-9
 - working with a particular version of, 2-13
- Comparison Viewer
 - function of, 1-10
- completed task
 - refreshing the display, 2-13
 - viewing, 2-13
- completing
 - pending task, 2-12
- conceptual help topics
 - accessing, 1-26
 - definition, 1-26
- Contents page for help
 - accessing, 1-25
- context-sensitive help
 - accessing, 1-25
- copying
 - change management objects to a different repository, 2-14
 - object definitions to a destination database, 1-11, 1-12, 1-14

- table definitions and data to a destination database, 1-15, 3-13
 - table definitions and data using drag and drop, 1-15
- Create Baseline application
 - capturing definitions in a baseline, 1-9
 - capturing definitions in SQL DDL format, 1-9
 - function of, 1-7
 - overview of, 1-8
- creating
 - a new version of a baseline, 2-8
 - a new version of a change plan, 2-10
 - a new version of a comparison, 2-9
 - baseline, 2-7
 - change plan, 2-10
 - comparison, 2-8
 - directive, 3-7
 - exemplar, 3-10
 - modified exemplar, 3-13
 - scoped directive, 3-8
 - standalone repository, 2-2

D

- DB Alter application
 - function of, 1-7
 - overview of, 1-14
- DB Propagate application
 - function of, 1-7
 - overview of, 1-14
- DB Quick Change application
 - function of, 1-7
 - overview of, 1-13
- defining
 - See creating
- deleting
 - a previous version of a baseline, 2-8
 - a previous version of a change plan, 2-10
 - a previous version of a comparison, 2-9
 - a task from the History page, 2-13
 - baseline, 2-7
 - change plan, 2-10
 - comparison, 2-9
- dependency object definition
 - adding to a plan, 3-12

- definition, 3-12
 - viewing for an exemplar, 3-12
- dependent object definition
 - adding to a plan, 3-12
 - definition, 3-12
 - viewing for an exemplar, 3-12
- destination database
 - executing a script at, 3-20
 - generating a script for, 3-15
 - keeping script changes made at, 3-20
 - selecting, 3-14
 - undoing script changes made at, 3-20
- directive
 - applications used to create, 3-6
 - defining, 3-7
 - definition, 3-4
 - extending the scope of, 3-5
 - viewing the changes specified by, 3-8
- drag and drop copying of object definitions, 1-15

E

- errors
 - generated during script execution, 3-20
 - generated during script generation, 3-18
- execute command
 - enabling logging for, A-11
- executing
 - a script, 1-23, 3-20
 - required privileges, 3-20
 - a script using the command line interface, A-10
- exemplar
 - actions performed at a destination database, 3-10
 - applications used to create, 3-6
 - defining, 3-10
 - definition, 3-5
 - modifying, 3-13
 - viewing attributes for, 3-11
 - viewing dependency object definitions for, 3-12
 - viewing dependent object definitions for, 3-12
 - viewing grants for, 3-11
- exit status codes
 - returned by command line interface, A-14
- exporting

baseline, 2-8, 2-14
change plan, 2-10, 2-14
comparison, 2-9, 2-14

F

failure prediction during script generation, 3-22

G

generate command
 enabling logging for, A-10
generating
 a script, 1-22, 3-15
 a script using the command line interface, A-9
glossary
 accessing, 1-26
grants
 excluding from an exemplar, 3-11
 including for an exemplar, 3-12
 viewing for an exemplar, 3-11

H

help
 accessing the glossary, 1-26
 displaying the Contents page, 1-25
 finding a particular type of topic, 1-26
 finding a specific topic, 1-25
 locating a conceptual or task topic, 1-26
 obtaining for a particular context, 1-25
 using, 1-25
history entries
 saving to a file, 2-12, 2-13
 sorting on the History page, 2-13
 sorting on the Pending page, 2-12
 viewing for a change plan, 2-11
 viewing for a comparison, 2-9
 viewing for a task, 2-11
History page
 manually moving a pending task to, 2-12
 removing a task, 2-13
 viewing new history entries for, 2-13

I

impact
 determining for a script prior to its
 execution, 3-18
importing
 baseline, 2-8, 2-14
 change plan, 2-10, 2-14
 comparison, 2-9, 2-14

J

job system
 using to execute a script, 3-21

K

keeping script changes, 3-20

L

log (trace) files
 location of, B-2
logging
 enabling during baseline creation with the
 capture command, A-7
 enabling during comparison creation with the
 compare command, A-8
 enabling during script execution with the execute
 command, A-11
 enabling during script generation with the
 generate command, A-10

M

mapping schemas, 3-15
modified exemplar
 applications used to create, 3-6
 creating with Plan Editor, 3-13
 definition, 1-2
modifying
 change plan, 2-10
 object definitions using Plan Editor, 3-2

O

object definitions

- basic steps for changing, 1-21
- capturing using Create Baseline, 1-8
- capturing using the command line interface, A-7
- changes that cannot be undone using the recovery script, 3-21
- comparing using the command line interface, A-8
- copying table definitions and data, 3-13
- copying using drag and drop, 1-15
- keeping changes made to, 3-20
- making changes to definitions in one or more databases, 1-14
- making one or more changes to one definition, 1-13
- modifying using Plan Editor, 3-2
- reproducing, 3-10
- reproducing one or more object definitions in one or more databases, 1-14
- specifying changes for, 3-7
- supported by change management applications, 1-3
- synchronizing one set with another, 1-11, 1-12
- undoing changes made to, 3-20
- viewing the changes specified for, 3-8

objects

- created by change management applications, 1-4
- obsolete script, 3-15
- OCM_FAILURE_PREDICTION property
 - enabling to predict script execution failures, 3-22
- omsconfig.properties file
 - description of properties, 3-22
 - location of, 3-22
 - used to enable script execution failure prediction, 3-22

P

pending task

- completing, 2-12

- manually moving to the History page, 2-12
- refreshing the display, 2-12
- saving history entries to a file, 2-12
- viewing, 2-12

plan

- See change plan
- Plan Editor application
 - modifying and creating object definitions using, 3-2
 - overview of, 1-16
 - right mouse button support for, 3-3
- predicting script execution failures, 3-22
- propagation options
 - viewing and modifying, 3-12

Q

Quick Tour

- starting, 1-18

R

recovery script

- attribute changes not undone by, 3-21

refreshing

- list of completed tasks, 2-13
- list of pending tasks, 2-12

removing

- See deleting

renaming

- change plan, 2-10

report

- about the dependency and dependent objects for an exemplar, 3-12
- about the impact of executing a script, 3-17
- about the objects in a baseline, 1-9
- about the results of a comparison, 1-11
- generating for completed task history entries, 2-13
- generating for pending task history entries, 2-12

repository

- copying change management objects to another, 2-14
- storing change management objects in, 2-1
- See also standalone repository

- reproducing
 - object definitions at a destination
 - database, 1-11, 1-12, 1-14
 - table definitions and data at a destination
 - database, 1-15, 3-13
 - table definitions and data using drag and drop, 1-15
- right mouse button support
 - for Change Manager, 2-6
 - for Plan Editor, 3-3
- rolling back script changes, 3-20

S

- scheduling
 - execution of command line interface
 - scripts, A-4
 - submission of change plan scripts to OEM job system, 1-3
- schema mapping, 3-15
- scoped directive
 - applications used to create, 3-6
 - creating, 3-8
 - definition, 3-5, 3-8
- scratch tablespace
 - specifying for script operations that require temporary data storage, 3-16
- script
 - dealing with problems in, 3-19
 - determining impact of prior to execution, 3-18
 - editing, 3-18
 - enabling failure prediction during generation, 3-22
 - executing, 3-20
 - privileges required, 3-20
 - executing using the command line interface, A-10
 - executing using the job system, 3-21
 - executing using the Oracle Enterprise Manager job system, 3-21
 - generating for a destination database, 3-15
 - generating for a destination database using the command line interface, A-9
 - keeping changes made by, 3-20
 - obsolete, 3-15

- resolving problems that occur during execution, 3-22
- scheduling using the command line interface, A-4
- stale, 3-22
- undoing changes made by, 3-20
- viewing a summary of, 3-18
- viewing the results after executing, 3-20

- sorting
 - history entries for completed tasks or manually moved pending tasks, 2-13
 - history entries for pending tasks, 2-12
- source database
 - choosing when creating change requests in Plan Editor, 3-4
 - definition, 3-4
- SQL statements
 - capturing definitions using Create Baseline, 1-9
- stale script, 3-22
- standalone repository
 - creating, 2-2
 - prerequisites for creating, 2-2
 - storing change management objects in, 2-2
 - using a different, 2-2
- starting
 - change management applications from Change Manager, 2-6
 - Change Manager, 2-3
 - Synchronization Wizard, 1-12
 - the Quick Tour, 1-18
- status codes
 - returned by command line interface, A-14
- supported database objects, 1-3
- Synchronization Wizard application
 - function of, 1-7
 - overview of, 1-12
 - starting, 1-12
- synchronizing
 - one set of object definitions with another, 1-11, 1-12

T

- task
 - completing pending, 2-12

- manually moving a pending task to the History page, 2-12
- removing from the History page, 2-13
- viewing completed, 2-13
- viewing history entries for, 2-11
- viewing new completed tasks, 2-13
- viewing pending, 2-12

task help topics

- accessing, 1-26
- definition, 1-26

trace files

- location of, B-2

tracking changes, 1-6

translator

- definition, 3-22

troubleshooting

- script execution errors, 3-22
- script execution failures, 3-22
- script generation errors, 3-18
- using the ClientConfig.properties file, B-2

U

- undoing script changes, 3-20

UNIX

- starting Change Manager, 2-3
- starting Change Manager from the command line, 2-3
- using command line interface on, A-1

updating

- list of completed tasks, 2-13
- list of pending tasks, 2-12

V

version

- baseline, 2-13
- change plan, 2-13
- comparison, 2-13

viewing

- a previous version of a baseline, 2-8
- a previous version of a change plan, 2-10
- a previous version of a comparison, 2-9
- baseline, 2-7
- change plan, 2-10

- comparison, 2-9
- completed task, 2-13
- history entries for a baseline, 2-8
- history entries for a change plan, 2-11
- history entries for a comparison, 2-9
- pending task, 2-12

W

Windows NT

- starting Change Manager, 2-3, 2-4
- starting Change Manager from DOS command line, 2-3
- using command line interface on, A-1

