

## **JD Edwards EnterpriseOne Tools**

Product Packaging Guide

Release 8.98 Update 4

**E14716-02**

March 2011

E14716-02

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	ix
Audience .....	ix
Documentation Accessibility .....	ix
Related Documents .....	ix
Conventions .....	x
 <b>1 Introduction to JD Edwards EnterpriseOne Product Packaging</b>	
1.1 JD Edwards EnterpriseOne Product Packaging Overview .....	1-1
1.2 JD Edwards EnterpriseOne Product Packaging Implementation .....	1-1
 <b>2 Understanding Product Packaging</b>	
2.1 Software Change Management .....	2-1
2.2 Distributed Development .....	2-2
2.3 Object Packaging .....	2-2
2.4 Software Mastering Tools .....	2-3
 <b>3 Working with Object Packaging</b>	
3.1 Using Object Packaging in JD Edwards EnterpriseOne Object Management Workbench .....	3-1
3.1.1 Understanding JD Edwards EnterpriseOne Object Management Workbench .....	3-1
3.1.2 Understanding Object Packaging .....	3-1
3.1.3 Saving an Object to a PAR File .....	3-2
3.1.4 Restoring an Object from a PAR File .....	3-2
3.2 Using Object Packaging in JD Edwards EnterpriseOne Configuration Assistant .....	3-3
3.2.1 Understanding JD Edwards EnterpriseOne Configuration Assistant .....	3-3
3.2.2 Including JD Edwards EnterpriseOne Objects in a Configuration .....	3-3
 <b>4 Working with Software Masters</b>	
4.1 Understanding Software Master Creation .....	4-1
4.2 Preparing for Software Master Creation .....	4-2
4.2.1 Setting Up the System TEMP Directory .....	4-2
4.2.2 Setting Up the UBEs .....	4-2
4.2.3 Creating Pristine Control Tables .....	4-3
4.2.4 Creating a Data Dictionary in the Pristine Data Source .....	4-4

4.2.5	Creating Change Tables.....	4-5
4.2.6	Verifying the Creation of Change Tables .....	4-7
4.3	Creating a Software Master .....	4-7
4.3.1	Understanding Software Master Definitions.....	4-8
4.3.1.1	Custom Data Inclusions in the Software Definition.....	4-8
4.3.1.2	Custom Table Inclusions .....	4-8
4.3.1.3	Custom Media Object Inclusions .....	4-8
4.3.1.4	Change Table Configuration.....	4-8
4.3.1.5	Final Master Creation.....	4-9
4.3.2	Prerequisite .....	4-9
4.3.3	Defining a Software Master.....	4-9
4.3.4	Configuring Change Tables .....	4-11
4.3.5	Setting the Package Build Application to the Mastering Mode.....	4-13
4.3.6	Validating a Software Master Definition.....	4-13
4.3.7	Creating a Final Software Master .....	4-14
4.3.8	Confirming the Software Master .....	4-15
4.3.9	Moving the Software Master to a CD ROM.....	4-15
4.3.10	Updating JD Edwards EnterpriseOne with the Software Master .....	4-16
4.4	Deleting a Software Master .....	4-19
4.4.1	Form Used to Delete Software Masters .....	4-19
4.5	Working with a Software Master Definition.....	4-19
4.5.1	Understanding Software Master Revisions .....	4-19
4.5.2	Understanding Software Master Validation.....	4-20
4.5.3	Forms Used to Revise Software Masters .....	4-20
4.5.4	Revising a Software Master Definition.....	4-20
4.5.4.1	CD Information .....	4-21
4.5.4.2	Build Steps .....	4-21
4.5.4.3	Directory Structure.....	4-22
4.5.4.4	Task List .....	4-22
4.5.5	Loading a Subdirectory from an Enterprise Network Directory.....	4-23
4.5.6	Copying a Software Master Definition.....	4-24
4.5.7	Proofing a Software Master Definition.....	4-24
4.6	Working with Software Master Templates .....	4-24
4.6.1	Understanding Software Master Templates .....	4-24
4.6.2	Understanding the All Build Step Processes .....	4-25
4.6.2.1	Undefined .....	4-25
4.6.2.2	Create CD File Structure.....	4-25
4.6.2.3	Package Build.....	4-25
4.6.2.4	Manual Build Step .....	4-25
4.6.2.5	Pre-Build Package.....	4-26
4.6.2.6	Pre-Build Pristine.....	4-26
4.6.2.7	Build Cross-Reference.....	4-26
4.6.2.8	Build Data Dictionary .....	4-26
4.6.2.9	Build In-House Package Database .....	4-26
4.6.2.10	Build Planner Database .....	4-26
4.6.2.11	Build Pristine Database.....	4-26
4.6.2.12	Build Store & Forward Database .....	4-26

4.6.2.13	Build Standalone Database .....	4-27
4.6.2.14	Build Change Tables .....	4-27
4.6.2.15	Master Central Objects .....	4-27
4.6.2.16	Compress .....	4-27
4.6.2.17	Create INF.....	4-27
4.6.2.18	Copy CD Directory Files .....	4-27
4.6.2.19	Virus Check .....	4-27
4.6.2.20	Windows Executable.....	4-28
4.6.2.21	Data Cleanup.....	4-28
4.6.2.22	Build Package Database.....	4-28
4.6.2.23	Batch Process .....	4-28
4.6.3	Understanding the ASU CD Template .....	4-28
4.7	Working with Custom CD Templates .....	4-30
4.7.1	Forms Used to Work with Software Master Templates .....	4-31

## 5 Working with Change Table Configurations

5.1	Understanding Change Table Revisions .....	5-1
5.2	Working with Change Table Configurations.....	5-1
5.2.1	Forms Used to Work with Change Tables .....	5-1
5.2.2	Setting Up Change Table Definitions .....	5-2
5.3	Working with Change Table Director.....	5-4
5.3.1	Forms Used to Work With Change Table Director.....	5-4
5.3.2	Adding a Change Table Definition .....	5-5
5.3.2.1	Change Table Director .....	5-5
5.3.2.2	Target Release Selection .....	5-6
5.3.2.3	Source Environment Selection .....	5-7
5.3.2.4	Batch Application Selection .....	5-9
5.3.2.5	Additional Change Table Definitions.....	5-10

## 6 Working with the Data Cleanup Procedure Application

6.1	Understanding the Data Cleanup Procedure Application .....	6-1
6.2	Working with the Data Cleanup Procedure Application .....	6-1
6.2.1	Forms Used to Work with Data Cleanup Procedures.....	6-1
6.2.2	Using the Data Cleanup Procedure Application.....	6-2
6.3	Adding a Data Cleanup Procedure .....	6-3
6.3.1	Forms Used to Add a Data Cleanup Procedure.....	6-3
6.3.2	Adding Data Cleanup Procedures .....	6-3
6.3.2.1	Procedure Definition Director .....	6-3
6.3.2.2	Add Procedure.....	6-3
6.3.2.3	Procedure Revision Information .....	6-5
6.3.2.4	Procedure Revision Steps .....	6-7
6.4	Copying a Data Cleanup Procedure .....	6-8
6.4.1	Forms Used to Copy a Data Cleanup Procedure .....	6-9
6.4.2	Copying a Data Cleanup Procedure .....	6-9
6.5	Executing a Data Cleanup Procedure and Build Step .....	6-9
6.5.1	Forms Used to Execute a Data Cleanup Procedure and Build Step.....	6-10

6.5.2	Executing a Data Cleanup Procedure and Build Step .....	6-10
-------	---	------

## 7 Working with Advanced Operations

7.1	Working with CD Templates .....	7-1
7.1.1	Understanding CD Templates .....	7-1
7.1.2	Forms Used with Mastering Templates .....	7-1
7.1.3	Setting Up CD Templates .....	7-2
7.1.4	Adding a New CD Template .....	7-4
7.1.4.1	Template Director .....	7-4
7.1.4.2	Template Information .....	7-5
7.1.4.3	CD Revisions .....	7-7
7.1.5	Copying CD Configuration .....	7-10
7.2	Converting Relative and Absolute Paths .....	7-12
7.2.1	Understanding Path Tokens.....	7-12
7.2.2	Forms Used to Work With Relative and Absolute Paths.....	7-13
7.2.3	Using Relative Path Tokens.....	7-13
7.2.4	Converting Relative and Absolute Paths .....	7-13
7.3	Working with Data Cleanup Procedure Templates.....	7-15
7.3.1	Forms Used with Data Cleanup Procedure Templates.....	7-15
7.3.2	Setting Up Data Cleanup Procedure Templates .....	7-15
7.3.3	Adding a Data Cleanup Procedure.....	7-16
7.3.3.1	Procedure Definition Director .....	7-16
7.3.3.2	Add Procedure.....	7-16
7.3.3.3	Procedure Revision .....	7-17
7.3.4	Copying a Data Cleanup Procedure .....	7-18
7.4	Viewing a Table Change Table .....	7-18
7.4.1	Forms Used to Work with Table Change Table .....	7-18
7.4.2	Viewing the Table Change Table.....	7-18
7.5	Viewing the Index Change Tables.....	7-20
7.5.1	Forms Used to Work with Index Change Table.....	7-21
7.5.2	Viewing the Index Change Tables.....	7-21
7.6	Viewing the Control Change Tables .....	7-23
7.6.1	Forms Used with Software Update Control Tables.....	7-24
7.6.2	Viewing the Software Update Control Tables.....	7-24

## 8 Working with Translation Updates

8.1	Understanding Translation Updates.....	8-1
8.2	Creating Translation Update Templates .....	8-1
8.2.1	Understanding Translation Update Templates.....	8-2
8.2.2	Prerequisites .....	8-2
8.2.3	Setting Processing Options for the Translation Update Template (R95160).....	8-2
8.2.3.1	Template Options .....	8-2
8.3	Collecting Translation Updates .....	8-2
8.3.1	Forms Used to Collect Translation Updates .....	8-2
8.3.2	Creating a Translation Updates Database.....	8-3
8.3.2.1	Work With Language Groups .....	8-3
8.3.2.2	Language Group Information.....	8-4

8.3.2.3	Work with OMW Project Detail .....	8-4
8.3.2.4	OMW Project Search & Select .....	8-4
8.3.2.5	Select Languages.....	8-8
8.4	Creating an Update Package .....	8-8
8.4.1	Understanding Package Updates.....	8-8
8.4.2	Prerequisites .....	8-8
8.4.3	Forms Used to Create an Update Package.....	8-9
8.4.4	Creating an Update Package.....	8-9
8.4.4.1	Work With Software Mastering.....	8-9
8.4.4.2	CD Configuration Director .....	8-10
8.4.4.3	CD Revisions .....	8-11

## **A Troubleshooting Product Packaging**

A.1	Failure During the Check Mastering Items Step .....	A-1
A.1.1	Solution .....	A-1
A.2	Failure to Build the Feature INF File.....	A-1
A.2.1	Solution .....	A-1
A.3	Failure to Copy Change Tables.....	A-1
A.3.1	Solution .....	A-2
A.4	Failure to Copy Custom Tables .....	A-2
A.4.1	Solution .....	A-2
A.5	Unable to Find Product Package After Installation .....	A-2
A.5.1	Solution .....	A-2
A.6	Problems with Object Librarian Records.....	A-2
A.6.1	Solution .....	A-2

## **Glossary**

## **Index**





---

# Preface

Welcome to the JD Edwards EnterpriseOne Tools Product Packaging Guide.

## Audience

This guide is intended for system administrators and technical consultants who are responsible for product packaging.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

## Related Documents

You can access related documents from the JD Edwards EnterpriseOne Release Documentation Overview pages on My Oracle Support. Access the main documentation overview page by searching for the document ID, which is 876932.1, or by using this link:

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=876932.1>

To navigate to this page from the My Oracle Support home page, click the Knowledge tab, and then click the Tools and Training menu, JD Edwards EnterpriseOne, Welcome Center, Release Information Overview.

This guide contains references to server configuration settings that JD Edwards EnterpriseOne stores in configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, and so on). Beginning with the JD Edwards EnterpriseOne Tools Release 8.97, it is highly recommended that you only access and manage these settings for the supported server types using the Server Manager program. See the Server Manager Guide on My Oracle Support.

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>Bold</b>	Indicates field values.
<i>Italics</i>	Indicates emphasis and JD Edwards EnterpriseOne or other book-length publication titles.
<code>Monospace</code>	Indicates a JD Edwards EnterpriseOne program, other code example, or URL.

---

# Introduction to JD Edwards EnterpriseOne Product Packaging

This chapter contains the following topics:

- [Section 1.1, "JD Edwards EnterpriseOne Product Packaging Overview"](#)
- [Section 1.2, "JD Edwards EnterpriseOne Product Packaging Implementation"](#)

## 1.1 JD Edwards EnterpriseOne Product Packaging Overview

Oracle's JD Edwards EnterpriseOne Product Packaging describes the different methods used to package JD Edwards EnterpriseOne objects and data so that they can easily be moved from one machine to another. You can use the Object Packaging Tools or Software Mastering Tools to package and move the objects and data.

The Object Packaging Tools provide a streamlined alternative to using the full set of Software Mastering Tools. There are two methods for using the Object Packaging Tools. The first method uses Object Management Workbench (OMW) to provide a simple way to transfer a single newly-developed or customized object from one machine to another in order to test it or to continue development. The second method uses Configuration Assistant to easily develop and package configurations and reuse the intellectual property that you have developed.

The full set of Software Mastering Tools gives you the ability to create your own software mastering processes. In this capacity, the tools oversee the entire mastering process from defining a software master to mastering a CD. The tools start with the definition process, where you provide as much initial information as possible. After you enter the critical information, the tools provide batch processes to validate the software mastering information before mastering, to master the software, and to validate the final software master image. This process gives you the ability to determine which objects should be shared between locations and create a custom installation that can be burned onto a CD. Then the CD can be used to deploy this custom installation anywhere within the enterprise.

## 1.2 JD Edwards EnterpriseOne Product Packaging Implementation

JD Edwards EnterpriseOne standardizes and automates software installation, making many steps transparent to users. Technical setup is preconfigured to meet the requirements of many JD Edwards EnterpriseOne customers. In addition, JD Edwards EnterpriseOne products are preintegrated and share a common database, which reduces the implementation process, minimizes ongoing administration, and provides customers with the flexibility to add new applications, modules, and tools as needed.



---

# Understanding Product Packaging

This chapter contains the following topics:

- [Section 2.1, "Software Change Management"](#)
- [Section 2.2, "Distributed Development"](#)
- [Section 2.3, "Object Packaging"](#)
- [Section 2.4, "Software Mastering Tools"](#)

## 2.1 Software Change Management

Prompted by the move to distributed client and server environments, software change management has assumed increasing importance. Customers want software that maximizes their return on investment. Development cycles are shrinking, and companies need to maintain a competitive advantage and speed up time to market. In addition, development environments today are more complex than ever. Organizations may have hundreds or even thousands of developers distributed around the world. It has never been more challenging to manage maximum developer productivity while ensuring high-quality and rapid deployment cycles.

Change management includes the following areas:

### **Version Control**

This encompasses the check-in and checkout or revision management of code, models, and so forth.

### **Build Management**

This represents the re-creation of an application for various revisions.

### **Promotion Management**

This is the control of logical and physical movement of code between development, testing, and production.

### **Release Management**

This is the notification and electronic distribution of software.

### **Problem Tracking**

This is the reporting, tracking, and resolution of application problems.

### **Configuration Management**

This encompasses the management of both physical level software components (source code) and logical level components (specification components, design diagrams, and so forth).

**Project Management**

This includes project planning and scheduling, task assignment, and tracking.

**Distributed Code Management**

This encompasses managing and synchronizing code from multiple geographic locations.

## 2.2 Distributed Development

Distributed development refers to the ability to perform development using one installation of JD Edwards EnterpriseOne and deploying that development to another installation.

For example, a developer in Paris could create new objects using the Paris installation of JD Edwards EnterpriseOne. An administrator in Paris could then use the JD Edwards EnterpriseOne Product Packaging Tools to create a CD that includes those new objects. This CD could be shipped to London where an administrator updates the London installation of JD Edwards EnterpriseOne (assuming that both installations are at the same release level). The JD Edwards EnterpriseOne Product Packaging Tools described in this guide provide this functionality.

This ability should not be confused with remote development, which is the creation of JD Edwards EnterpriseOne objects by a developer who is completely disconnected from any installation. With remote development, the developer can connect to a JD Edwards EnterpriseOne installation later and merge those objects.

JD Edwards EnterpriseOne Product Packaging helps manage the complex development environment by enabling you to synchronize development teams across remote locations or multiple platforms.

JD Edwards EnterpriseOne Product Packaging can assist in overcoming some of the challenges of distributed development such as:

**Multiple Physical Locations**

Increasingly, development organizations are located across multiple physical locations, from adjacent buildings to across continents.

**Multiple Platforms**

Not only may your developers be in different physical locations, they may also be developing on and for multiple platforms.

**Outsourcing and Contractors**

In addition, the use of outsourcing and subcontractors has exploded in the software industry. A challenge for many organizations involves tracking the software source code that third-party software vendors provide. The software may be being built not only on Oracle software, but also on vendor software.

**Localizations**

Finally, things become even more complex when you need to make local changes to the supplied software. All of these factors create an urgent need for management of software across multiple geographically dispersed locations.

## 2.3 Object Packaging

Object packaging provides a very simple way to move any JD Edwards EnterpriseOne object from one machine to another through the creation of a zip file that contains compressed files and directories. The zip file, called a PAR file, can consist of JD

Edwards EnterpriseOne objects from either local specifications or from central objects. Depending on the purpose of the PAR file, it may also contain:

- JD Edwards EnterpriseOne objects in an XML metadata specification file format.
- Data dictionary items, user-defined codes, and new tables in XML file format.
- Deployment components and a schedule of tasks to be performed, as well as documentation on these components and tasks.
- Oracle Business Accelerator (OBA) files such as the Business Accelerator Schedule file, help files, and business process documentation.
- Data PAR files that contain base configuration data for OBA.

Use object packaging within Object Management Workbench to create PAR files from local specifications. Use object packaging within Configuration Assistant to create PAR files from central objects. Once a PAR file is created, it can be easily transported from one machine to another (for example, through email or a USB flash drive).

## 2.4 Software Mastering Tools

Oracle's JD Edwards EnterpriseOne Software Mastering Tools are used to perform these basic processes for JD Edwards EnterpriseOne Product Packaging:

- Package definition.
- Batch process to validate the definition.
- Batch process to master the software.
- Batch process to validate the master image.

The tools used for mastering are:

### **Software Mastering Director (P9640)**

Oracle's JD Edwards EnterpriseOne Software Mastering Director automates many routine, repetitive, and time-consuming mastering tasks, reducing manual processes and increasing control. In addition, verification and master reports inform you about each step of the process.

The JD Edwards EnterpriseOne Software Mastering Director application:

- Provides the ability to build a mastered version of objects or modules.
- Uses a series of batch processes (UBEs) and manual tasks that build all of the components of a master.
- Creates a final product that is an image master in a tree format, which administrators can burn onto a CD.
- Provides verification and master status reports.

### **Change Table Configuration Director (P9642)**

Oracle's JD Edwards EnterpriseOne Change Table Configuration Director enables you to share code across all or some of the locations. JD Edwards EnterpriseOne keeps an inventory of code and data changes in files known as change tables. The JD Edwards EnterpriseOne Change Table Configuration Director provides a JD Edwards EnterpriseOne administrator with the ability to define how a module's change tables are to be built. You can even use the Director to define a source set of changes and move them to multiple target development locations.

The JD Edwards EnterpriseOne Change Table Configuration Director:

- Defines which change tables are to be included.
- Specifies multiple target and source release combinations.
- Runs change tables from a central master batch process (UBE).

**Data Cleanup Procedure Application (P9646)**

Use Oracle's JD Edwards EnterpriseOne Data Cleanup application to create procedures and templates to clean up the software master data. The data cleanup application can delete obsolete data, standardize existing data, and provide integrity, summary, and error reports about the data. You can clean up these items:

- Data Dictionary tables and specifications.
- Autopilot tables.
- User-defined codes and types.
- Menu tables.
- Object Librarian tables.
- Central Objects tables.



---

## Working with Object Packaging

This chapter contains the following topics:

- [Section 3.1, "Using Object Packaging in JD Edwards EnterpriseOne Object Management Workbench"](#)
- [Section 3.2, "Using Object Packaging in JD Edwards EnterpriseOne Configuration Assistant"](#)

### 3.1 Using Object Packaging in JD Edwards EnterpriseOne Object Management Workbench

This section provides overviews of JD Edwards EnterpriseOne Object Management Workbench and object packaging and discusses how to:

- Save an object to a PAR file.
- Restore an object from a PAR file.

#### 3.1.1 Understanding JD Edwards EnterpriseOne Object Management Workbench

JD Edwards EnterpriseOne Object Management Workbench (OMW) is the primary component of the change management system for EnterpriseOne development. A change management system is vital to a productive development environment because it helps organize a myriad of development activities and helps prevent problems, such as when a developer intermixes components from different releases or when multiple developers simultaneously change an object. JD Edwards EnterpriseOne OMW automates many of these change management activities.

OMW provides a way for developers to save and restore an entire OMW project or individual JD Edwards EnterpriseOne objects from their local specifications. This provides a simple way to backup development work or move objects between machines. For example, a developer may need to pass their JD Edwards EnterpriseOne objects to a quality engineer for testing.

See *JD Edwards EnterpriseOne Tools Object Management Workbench Guide*.

#### 3.1.2 Understanding Object Packaging

Object packaging within OMW is the mechanism by which you can save and restore your OMW project or JD Edwards EnterpriseOne objects from and to local specifications. This is done by using a PAR file. A PAR file is similar to a zip file in that it contains compressed files and directories. Developers can use OMW to create PAR files for individual objects in order to easily transfer the object to another machine for

testing or further development. JD Edwards EnterpriseOne objects are stored in XML format within the PAR files.

### 3.1.3 Saving an Object to a PAR File

To save an object to a PAR file, access JD Edwards EnterpriseOne OMW.

1. In JD Edwards EnterpriseOne OMW, select the object or project that you want to save.
2. Click the Save button in the center column.

The jdeomw dialog box appears. OMW provides two methods with which to save an object. The first method is to save the object to a zip file. If you save the object to a zip file, OMW saves the object or project as a local copy in a PAR file format, which is compatible with Zip, Jar, and JD Edwards Change Assistant technologies. The second method is to save the object to a shared database location.

3. Click Yes to save the object to a zip file.
4. On the Select Folder form, search for and select the folder to which you want to save the object, and then click Open.

OMW creates a PAR file for your object at the specified location. You can copy the PAR file to email or portable storage and easily move it to another machine.

### 3.1.4 Restoring an Object from a PAR File

To restore an object from a PAR file, access JD Edwards EnterpriseOne OMW.

1. In JD Edwards EnterpriseOne OMW, select the object or project that you want to restore.
2. Click the Restore button in the center column.
3. On the Overwrite/Delete Specs Confirmation window, click Yes.

The jdeomw dialog box appears and provides options for you to choose from where you want to restore the object. You can choose to restore the object from a zip file, or to restore the object from the Save location specified by the OMW administrator.

4. Click Yes to restore the object from a zip file.

---

**Note:** If you have saved at the object level, you will restore at the object level. If you have saved at the project level, you will restore at the project level. To restore individual objects from the project, you can use WinZip to open the PAR file and extract the individual objects.

---

5. On the Select Folder form, search for and select the PAR file from which you want to restore the object or project, and then click Open.

OMW restores the object or project. For objects that are new, all appropriate information is created. All actions are logged by the system.

---

**Note:** You can use OMW Allowed Actions to limit access to the save/restore functionality if necessary.

---

## 3.2 Using Object Packaging in JD Edwards EnterpriseOne Configuration Assistant

This section provides an overview of JD Edwards EnterpriseOne Configuration Assistant and discusses how to include JD Edwards EnterpriseOne objects in a configuration.

### 3.2.1 Understanding JD Edwards EnterpriseOne Configuration Assistant

The Configuration Assistant is a standalone Java-based tool provided by Oracle that allows you to extract a subset of JD Edwards EnterpriseOne configuration information into a portable archive. This data can then be loaded into any other JD Edwards EnterpriseOne environment using Change Assistant.

The Configuration Assistant provides another simple method to package JD Edwards EnterpriseOne objects as well as data and other configuration items. The Configuration Assistant is used with Oracle Business Accelerators (OBA) to assist you with the setup and configuration of business data, program versions, task views, user-defined codes (UDCs), and JD Edwards EnterpriseOne objects. Using Configuration Assistant, you can specify the configuration elements that you want to package, including objects, tables, versions, UDCs, security records, and task views. You can also specify the loading instructions for each configuration element into the target environment, which include whether to replace tables, replace records, or append records.

#### See Also:

- Online help within the Configuration Assistant application.
- *JD Edwards EnterpriseOne Tools Release 8.98 Configuration Assistant User Guide.*
- *JD Edwards EnterpriseOne Configuration Development Methodology 9.0 Guide.*
- "Using Change Assistant" in the *JD Edwards EnterpriseOne Tools 8.98 Software Update Guide.*

### 3.2.2 Including JD Edwards EnterpriseOne Objects in a Configuration

To include JD Edwards EnterpriseOne objects in a configuration:

---

**Caution:** This is a very basic example of how to create a configuration. Creating configurations requires more planning, development, and testing than discussed here. Please see *JD Edwards EnterpriseOne Configuration Development Methodology 9.0 Guide* for more information about configuration development.

---

1. Within Change Assistant, select Tools, Configuration Assistant.
2. In Configuration Assistant, select an existing configuration or add a new configuration.
3. Use the Search tab to find OMW projects and objects.
4. Select the projects and objects that you want to include in the configuration.
5. Select other items, such as tables and UDCs, that you want to include in the configuration.

6. Click the Extract Configuration Data button.

The Extract Status window appears and displays the status of the extraction as it occurs.

7. Click the Report button or look at the status bar to see if the extraction was successful.

A PAR file containing the selected objects is created and placed in your configuration directory.

---

---

**Note:** The configuration directory is defined within the Preferences for Configuration Assistant.

---

---

8. Use Change Assistant to restore the PAR file.

**See Also:**

- Online help within the Configuration Assistant application.
- *JD Edwards EnterpriseOne Configuration Development Methodology 9.0 Guide.*
- *JD Edwards EnterpriseOne Tools Release 8.98 Configuration Assistant User Guide.*

---

## Working with Software Masters

This chapter contains the following topics:

- [Section 4.1, "Understanding Software Master Creation"](#)
- [Section 4.2, "Preparing for Software Master Creation"](#)
- [Section 4.3, "Creating a Software Master"](#)
- [Section 4.4, "Deleting a Software Master"](#)
- [Section 4.5, "Working with a Software Master Definition"](#)
- [Section 4.6, "Working with Software Master Templates"](#)
- [Section 4.7, "Working with Custom CD Templates"](#)

### 4.1 Understanding Software Master Creation

This section discusses the software master creation process.

Use this checklist to ensure that you complete all of the processes that are necessary for creating a software master:

- Verify that at least 1 GB of space is available for the TEMP directory of the workstation from which you use JD Edwards EnterpriseOne Product Packaging Tools.

This memory requirement is the most memory that you might need for a software master. The actual size of the master depends on the size of the package that you create.

- Complete the tasks in this section.
- Define a software master using Oracle's JD Edwards EnterpriseOne CD Configuration Director.
- Provide information about the type of master that you want to create, such as its name, release number, and so on.
- Select the package that you want built into the master.
- Configure change tables using Oracle's JD Edwards EnterpriseOne Change Table Director.
- Provide the target release number for the change tables.
- Provide source environment information about the change tables.
- Select the change tables and their versions.
- Validate the software master definition.

- Create the final software master, which automatically validates the master.
- Use the software master to update JD Edwards EnterpriseOne.

## 4.2 Preparing for Software Master Creation

This section discusses how to:

- Set up the system TEMP directory.
- Set up the UBEs.
- Create pristine control tables.
- Create a data dictionary in the pristine data source.
- Create change tables.
- Verify the creation of change tables.

### 4.2.1 Setting Up the System TEMP Directory

This task creates a temporary directory that will be used to store the Oracle-supported local database for mastering the JD Edwards EnterpriseOne Product Packaging CD.

To set up the System TEMP directory:

1. From the Windows desktop, select Start, Settings, Control Panel.
2. On Windows Control Panel, double-click the System button.
3. On System Properties, click the Advanced tab.
4. Double-click Environment Variables.
5. Set the TEMP and TMP variables for both the User and System settings so that they all access the same directory (for example, c:\TEMP).

---



---

**Note:** Ensure that the windows temporary directory on the workstation has at least 1 GB free disk space available (the more space the better). If the temp directory has less than 1 GB of free space, you can create a temp directory on another disk drive with adequate space and set the system environment variables TEMP and TMP to point to the new temp folder. Reboot the system for the change to take effect.

---



---

This directory will store the software master that you create in the following tasks.

6. In the temporary directory that you specified in the previous step, create an empty subdirectory called \master.

---



---

**Note:** The name of this subdirectory is case-sensitive and should be all lowercase.

---



---

### 4.2.2 Setting Up the UBEs

JD Edwards EnterpriseOne is initially configured to accommodate internal Oracle naming conventions that are not relevant to customer installations. Therefore, customers must change the configuration of data source names and paths in the UBEs so that they correspond to the naming conventions that are described in this document.

---

**Note:** Unless you check out the versions before making the following changes, these changes will be available only on this workstation until another package is installed.

---

To set up the UBEs:

1. On the Mastering Workstation, enter **BV** in the FastPath field to open up Batch Version.
2. Search for the R9640E UBE, select the XJDE0002 version, and change the Component File Specifications parameter to \*.\*.
3. For each of the UBEs listed, search for the UBE, select the specified version, and change the following processing option for the "Data Source Name" to **OneWorld - Product Packaging**.

UBE	Version
R9670	XJDE0001
R9670	XJDE0002
R9670	XJDE0003
R96700	XJDE0001
R9600400C	XJDE0001

4. Create a JD Edwards EnterpriseOne Object Configuration Manager (OCM) mapping to run R9640B locally.

### 4.2.3 Creating Pristine Control Tables

For control table changes to be included on an update CD, the JD Edwards EnterpriseOne Product Packaging Tool requires a pristine copy of the control tables. If the installation includes the PS900 pristine environment, these tables already exist in the JD Edwards EnterpriseOne instance and you can bypass the following task. If the installation does not include the pristine environment, you can create a set of pristine control tables by copying the Control Table tables from Control Tables Local to a new data source: Control Tables - PS900. You perform this process only once.

To create pristine control tables:

1. On the Deployment Server, sign in to JD Edwards EnterpriseOne in the DEP900 environment.
2. From menu GH9011, select Batch Versions.
3. Type **R98403** in the Batch Application field and click Find.
4. On Work With Batch Versions - Available Versions, select version XJDE0501 (Control Tables For Pristine Database), and select Processing Options from the Row menu.
5. Select the Environment tab and change these processing options:
  - Target Environment (option 1)  
Leave this field blank.
  - Target data source (option 2)

Enter **Control Tables - PS900**.

- Data Load (option 3)

Enter **2** to copy data to the table.

- Source data source (option 4)

Enter **Control Tables Local**.

6. Run the version locally.

The report copies the F9000, F9001, F9002, F0004, F0004D, F0005, F0005D, F9005, F9005D, F9006, F9006D, F91100, F91100D, F91400, F91410, F91420, F91430, F91500, F91510, and F951000 tables from the Control Tables Local data source to Control Tables - PS900.

7. Verify the results of the report and the creation of the tables.

8. Quit the Batch Versions application.

9. From menu GH9011, select Object Configuration Manager.

10. On Machine Search and Select, select the machine with the System - 900 data source.

11. For the PS900 environment, add Object Configuration Manager mappings for tables F9000, F9001, F9002, F0004, F0004D, F0005, F0005D, F9005, F9005D, F9006, F9006D, F91100, F91100D, F91400, F91410, F91420, F91430, F91500, F91510, and F951000, and assign these new tables to the Control Tables - PS900 data source.

12. Activate the new mappings.

13. Quit Object Configuration Manager.

## 4.2.4 Creating a Data Dictionary in the Pristine Data Source

You must create a data dictionary in the pristine data source.

To create a data dictionary in the pristine data source:

1. From the deployment server, sign in to the JDEPLAN environment.

Use **JDE** as the user name and **JDE** as the password.

2. From the System Administration Tools menu (GH9011), select Batch Versions.

3. On Work With Batch Versions - Available Versions, type **R98403** into the Batch Application field and click Find.

4. Select version XJDE0509 and click Copy.

5. On Version Copy, enter a new version name and a new version title, and then click OK.

6. On Work With Batch Versions - Available Versions, select the version that you just created and select Processing Options from the Row menu.

7. Complete these processing options as indicated:

- Target data source (option 2)

**DATA DICTIONARY - PS900**

- Source environment (option 5)

**Data Dictionary Local**



Use these processing options to set up the batch process to copy the data dictionary tables from the JDEPLAN environment to the new pristine data dictionary.

8. Set the version that you just created to run in the proof mode first.
9. Run this report locally, not on the enterprise server.
10. Verify the results of this report.
11. When satisfied with the proof mode results, run the version that you just created in final mode.
12. Run OCM from the Fast Path field and select the enterprise server.
13. For the PS900 environment, add OCM mappings for F9200, F9202, F9203, F9207, F9210, F9211, and F9212 that point these tables to Data Dictionary - PS900.
14. Activate the new OCM mappings.

## 4.2.5 Creating Change Tables

JD Edwards EnterpriseOne Product Packaging requires that a set of tables, called change tables, exist and be mapped to the Control Tables - <environment> data source. A change table is a table that contains the items that have changed between releases or updates. Normally, these tables are generated during the installation process, when you run the environment workbench. If, however, the environment is populated with demo data, these tables will not be generated even though the environment workbench reports that they were created successfully.

Complete this task and the next if you need to create change tables in the mastering environment.

To create change tables:

1. On the deployment server, sign in to the JDEPLAN environment.
2. From menu GH9011, select Batch Versions.
3. Type **R98403** in the Batch Application field and click Find.
4. Select version XJDE0507 (Change Table) and click Copy.
5. On Version Copy, complete these fields as directed:
  - New Version
  - Version Title

Type a name for the new version.

CREATECT
6. Click OK.
7. On Batch Version Design, click the Processing Options button on the General tab.
8. Edit these processing options for CREATECT as directed:
  - Target Environment (option 1)  
Type the name of the mastering environment.
  - Target Data Source (option 2)  
Leave this field blank.
  - Data Load (option 3)

1: Load Production to create the table without data.

- Source data source (option 4)

**Control Tables Local**

9. Click OK.
10. On Batch Version Design, click Run.
11. On Version Prompting, select Data Selection and click Submit.
12. On Data Selection, verify that these tables are listed in the right operand for BC OBNM, and add those that are missing:
  - F960004
  - F960005
  - F967611
  - F969000
  - F969001
  - F969002
  - F969005
  - F969006
  - F9691100
  - F9691400
  - F9691410
  - F9691420
  - F9691430
  - F9691500
  - F9691510
  - F9695100
  - F9698710
  - F9698712
  - F9755
  - F9757
  - F9759
  - F9760
  - F98800DN
  - F98800N
  - F98800TN
  - F98810DN
  - F98810N
  - F98811N
  - F98830N

- F98840N
- F98845N

**13. Run the version.**

The program creates the change tables in Control Tables - CRP or whichever environment Control Table data source you specified.

## 4.2.6 Verifying the Creation of Change Tables

For each of the change tables, verify that an OCM mapping exists and points to the appropriate data source for the mastering environment.

To verify the creation of change tables:

1. From menu GH9011, select Object Configuration Manager.
2. On Machine Search and Select, select the machine with the System - 900 data source.
3. For the mastering environment, verify that each of these change tables is mapped to the correct data source and that each mapping is activated:

Change Table	Data Source
F960004, F960005, F969000, F969001, F969002, F969005, F969006, F9695100, F98800DN, F98800N, F98800TN, F98810DN, F98811N, F98810N, F98830N, F98840N, F98845N	Control Tables - <environment>
F9698710, F9698712	Central Objects - <environment>
F967611, F9691100, F9691400, F9691410, F9691420, F9691430, F9691500, F9691510,	System - 900
F9755, F9757, F9759, F9760	Data Dictionary - 900

4. Quit OCM.
5. Run Oracle's JD Edwards EnterpriseOne Universal Table Browser (UTB) to verify that the tables were created in the correct location.

## 4.3 Creating a Software Master

This section provides an overview of software master definitions, lists a prerequisite, and discusses how to:

- Define a software master.
- Configure change tables.
- Set the package build application to the mastering mode.
- Validate a software master definition.
- Create a final software master.
- Confirm the software master.
- Move the software master to a CD ROM.
- Update JD Edwards EnterpriseOne with the software master.

## 4.3.1 Understanding Software Master Definitions

A software master definition provides the structure for the finished software master. It establishes what template and packages to use, what build steps to follow, and the directory structure of the final master. The software master definition is only the structure or outline of the master. After you set up the software master definition, you validate it and then create the actual software master.

### 4.3.1.1 Custom Data Inclusions in the Software Definition

The JD Edwards EnterpriseOne Product Packaging application enables you to include custom data on the software master. Read the following tips for including custom tables and custom media objects on the software image.

### 4.3.1.2 Custom Table Inclusions

If the customer creates a new, customized table that is to be delivered with pristine data, the mastering process should automatically handle this procedure if these conditions are met:

- The table is included in the update package.
- The table is in the environment of the update package.
- The table is not included in the target environment.
- The table and index change table exist.

### 4.3.1.3 Custom Media Object Inclusions

To include custom media objects in the product package, add these build steps to the ASU template:

1. Add a build step to copy table records from the F00165 to the XML database.
2. Add a build step to copy the media object files from the deployment server to the image path directory.

### 4.3.1.4 Change Table Configuration

This example illustrates why change table configurations are needed:

Create change table configurations to indicate to the JD Edwards EnterpriseOne Product Packaging Tools the change tables that you want included with the software master. When you install the finished software master, the installation process will update the control tables of the enterprise with the change tables that you indicate.

Assume that the home office in Denver, USA, makes software changes and wants to update an enterprise that is not networked to the Denver enterprise. The other enterprise, which has its own set of JD Edwards EnterpriseOne software and control tables, is in Paris, France. The Denver enterprise creates a software master that includes the package containing the software changes, as well as the change tables that contain data dictionary and user-defined codes changes, which, for this example, are the only control table changes since the Paris enterprise was last updated. The Denver enterprise writes the software master to a CD and ships it to Paris. When the Paris enterprise installs the software master, the change tables update the Paris control tables, making the tables concurrent with the Denver control tables.

#### 4.3.1.5 Final Master Creation

This task explains how to create the final software master image from a definition that you previously created. You can master a single definition or master all definitions of a specific JD Edwards EnterpriseOne release.

The report that accomplishes this task is Oracle's JD Edwards EnterpriseOne Master CD (R9640B) report. This report controls the execution of all of the build steps and the UBEs that are associated with the build steps. This report also changes the status of each build step from **30 - Validated** to a new status of **60 - Built** or **50 - Failed**.

With an average workstation, network, and ERP environment, the mastering process should take about one hour. However, the software includes many variables that can affect the time that is necessary to build a product package CD. These variables are:

- Status and level of debugging.
- Number of objects in the update package.
- Size of the objects in the update package.
- Overall size of the parent package.
- Number of change tables to define.
- Network speed.
- Type of database.
- Build machine speed.
- Build machine memory.
- Number of other processes running on the enterprise server, deployment server, and build machine.

#### 4.3.2 Prerequisite

Ensure that at least 1 GB of space is available for the TEMP directory of the workstation from which you use the JD Edwards EnterpriseOne Product Packaging Tools. This memory requirement is the most memory that you might need for a software master. The actual size of the master depends upon the size of the package that you create.

**See Also:**

- [Working with Software Master Templates.](#)

#### 4.3.3 Defining a Software Master

To define a software master:

1. Sign in to a JD Edwards EnterpriseOne workstation using the development environment where the modifications exist, such as DV900.
2. From Product Packaging (GH962), select Product Packages (P9640).
3. On Work With Software Mastering, click Add.
4. On CD Configuration Director, click Next.
5. On CD Information, complete these fields:
  - Name

Enter a unique name for the software master definition that you are adding, such as **E900UPDATE**. This name must exactly match the name of the package to be included, and it must be in all uppercase letters.

- Release

Enter the release for the software master definition, such as **E900**. This is the release that is currently installed on the client workstation that you are using.

- Type

Specify the type of CD that you want to master. In this case, it is an ASU CD (option **09**) which is the default. If you select an option other than **09**, the software displays an error message when you press **Next**. This field indicates which software template to use when you further define the master. Templates include the build steps and directory structure that is necessary to create a software master.

- Build Phase

Enter the development phase that the software master definition represents, such as **alpha**, **beta**, or **general availability**.

- Description

(Optional) Enter a description of the software master definition.

- Image Path

Enter the directory path where you want the software master definition to reside on the JD Edwards EnterpriseOne Product Packaging build machine (for example, **d:\E900\PP**). The directory that you indicate will be the root directory for the software master that you create. It will contain the contents of the master. After you create the image, the CD can be burned through the CD burner software.

6. Click Next.

If you chose a template that appears on the list of values when you click the Search button for the Type field, but the template is not on the enterprise, the CD Template Does Not Exist form appears. You have probably entered the incorrect release number. Click OK to continue adding the software master definition, or click Cancel to stop adding the definition.

If the template that you chose using the Type field was set up to include packages, the CD Packages form appears.

7. Select the update package that includes the objects for the update disk.

You can either select an existing update package or create a new one. To ensure that the update package is defined correctly, note that:

- You can use only update packages for this process.
- The package name must be the same name as the product package that was previously defined, and it must be in all uppercase letters.
- The package should be created from its very beginning.
- The update package must always include specifications, build specifications, build business functions, and compression.

If the package is not compressed, then product packaging will not work correctly because JD Edwards EnterpriseOne Product Packaging looks only for cab files to copy to the image.

8. If the package is undefined, the Select Package Build form automatically appears; otherwise, click Select Package.

You can also add packages by clicking New Package Build or Package Assembly.

9. On Select Package Build, if you need to assemble or define a package or both, select one of these from the Form menu:
  - Pkg Assembly
  - Pkg Build

---

**Note:** Make sure that you compress any packages that you build.

---

See [Setting the Package Build Application to the Mastering Mode](#).

10. On the Select Package Build form, find and select a package, and then click Select.
11. On CD Packages, select the package marker that appears.

For an ASU CD, only one package is required. For other types of software masters, more than one package might be required.

12. Click Next.

If the template that you chose was set up to include change table configurations, after you click Next, the JD Edwards EnterpriseOne Change Table Director form appears.

13. Continue to the next task, which explains how to configure the change tables.

### 4.3.4 Configuring Change Tables

To configure change tables:

1. On the JD Edwards EnterpriseOne Change Table Director, which appears automatically during the mastering process if required by the template, click Next.  
  
Alternatively, from Product Packaging (GH962), select Change Table Configurations (P9642) and then click Add.
2. On the JD Edwards EnterpriseOne Change Table Director, click Next.
3. On the Target Release Selection form, complete the Target Release field.

Change the target release to a custom user-defined code (UDC) name that you are mastering. This name needs to be unique, such as E900TAX for the 1099 tax update. Select the release name using the visual assist button.

---

**Note:** The software will not function correctly if these tables have the same value for the source and target release. To add the custom UDC, press the visual assist and then click Revisions. This action displays the Work with User Defined Codes form. Work with User Defined Codes, click Add. Scroll to the bottom of the grid and add the new value in the last line of the grid. After you have created the UDC, select the new UDC on the Work with User Defined Codes form, and click Select. This action will populate the Target Release field with the new value.

---

4. Click Next.

5. On Source Environment Selection, complete these fields:

- Source Environment

Enter the name of the environment that includes the control tables that are used as the baseline for the changes. The source is typically the pristine environment (**PS900**). The software builds the change table records by comparing the target environment to this baseline.

- Source Release

Verify that the source release matches the release and the cumulative update level, such as **E900**, of the source environment that you want. The software automatically populates this field based upon the source environment.

6. Click Next.

The Batch Application Selection form appears. This form lists the available change tables along with the batch application and version that will create the change table.

7. To select the default change tables that you want with the software master, double-click the gray button to the left of the change table row.

A check mark appears on the button.

Select any or all of these change tables where you made additions, deletions, or changes. The batch process compares the source and target tables in each selected category and creates a change table that contains all changes.

- Data Dictionary
- User Defined Codes

This batch application should always be chosen because changes always occur to the UDC that defines the release.

- Workflow.
- Favorites.
- Templates and Smart Fields.
- Tips of the Day.
- Table and Index Changes.
- Solution Explorer.
- Service Constants. (8.97 only)

You can double-click a check mark to remove it.

8. To select a different version or to select multiple versions, select a change table row, and then select Version Selection from the Row menu.

9. On Version Selection, select one or more versions and click Select or double-click the gray button to the left of a version.

A check mark appears to the left of the versions that you chose.

10. After choosing the versions that you want, click Close.

The Batch Application Selection form reappears.

A check mark appears to the left of the change table row that you chose. The change tables that you want to be built for the software master must have check marks next to them or the software will ignore them. If you chose more than one



version, the word *<MULTIPLE>* appears under the Version column for that change table.

11. Either select another change table row and select Version Selection from the Row menu, or click Next.

If you click Next, the Additional Change Table Definitions form appears.

12. On Additional Change Table Definitions, perform one of these actions:

- To define additional change table configurations for a new source and target release combination, select Continue and click OK.

The program displays the Target Release Selection form. Return to the beginning of this task and repeat the steps for a different source and target combination.

- Select Quit and click OK.

This action stops the change table configuration director and displays the final revisions screen for mastering the defined CD.

If you select Quit, the CD Revisions form appears. Use this form to revise the software master definition.

13. On CD Revisions, click OK.

the software master definition is saved, and the CD Revisions form closes.

14. Continue to the next task, which explains how to validate the definition that you just created.

### 4.3.5 Setting the Package Build Application to the Mastering Mode

To compress an update package, you must set the package build application to the mastering mode.

To set Oracle's JD Edwards EnterpriseOne Package Build Application to the mastering mode:

1. On the Deployment server, log on to the DEP900 environment.
2. From the Package and Deployment Tools menu (GH9083), right-click Package Build, and select Prompt For Values.
3. On Processing Options, type 1 in option 2 to set up the mastering mode.
4. Click OK.
5. Quit the package build application.

### 4.3.6 Validating a Software Master Definition

This task explains how to run Oracle's JD Edwards EnterpriseOne Pre-Mastering CD Validation report (R9640A) to validate the software master definition that you set up and what to review in the report to help with the validation. The report runs in final mode. The report lists whether the build step is validated. The report will change the status of build steps that do *not* have errors.

To validate a software master definition:

1. From the Product Packaging menu (GH962), select Product Packages (P9640).
2. On Work With Software Mastering, click Find.

The existing software master definitions appear.

3. Select the master that you want to validate, and then select Validate from the Row menu.
4. On Report Output Destination, select to view the report on screen and click OK.

The report runs. All steps without errors whose statuses were In Definition (10) or Defined (20) are promoted to the status of Validated (30); they will work properly during the software mastering process. Otherwise, the build step will have either a warning or an error message. Error messages will stop the creation of a software master. You must fix any build steps that contain error messages. The message specifies the error.

Warning messages will not stop the creation of a software master. If a build step has a warning message, you should verify that the build step is as you want it before proceeding.

---

**Note:** The report will always include a warning stating that the build step is different from the template. This warning is always true because the process uses a custom package build.

---

5. Continue to the next task, which explains how to create the final software master.

### 4.3.7 Creating a Final Software Master

To create a final software master:

1. From Product Packaging (GH962), select Product Packages (P9640).
2. On Work With Software Mastering, perform one of these actions:
  - To create a software master from one specific definition, click Find, select the definition that you want to create, and then select Master CD from the Row menu.
  - To create software masters for all definitions of a specific release, select Master Release from the Row menu.

A form appears in which you enter the release number of the definitions that you want to master.

Either choice runs a batch process (R9640B) that creates the software master based upon the definition that you chose. Run the report locally and select the option to display the results on screen. Depending on the build steps that are necessary to complete, the report launches several other reports, each corresponding to a particular build step.

The software mastering process stops when it encounters a manual build step.

3. To complete the manual build step, select that build step, and select Execute Step from the View menu.

The status of that step is advanced to **60 (Built)**. Click Master CD or Master Release to continue the software master process. The progress of the build can be monitored from the CD Revisions screen.

4. From the View menu, select Refresh.

As each step is processed and completed, the icons change color and style. The last build step automatically validates the software master and generates a final report (R9840B) This report includes a comprehensive status of the product package build.

5. Review this report for errors.

If the validation report does not indicate any errors, you have successfully created a software master. This report validates that the number of directories match between the software master definition and the final software master. It also provides details about the file counts and the total size of the master in megabytes.

This process automatically creates a self-extracting executable file and a CAB file and places them in the \$TEMP/final directory using the name of the software master followed by the .exe and .cab extensions. Use the executable (.exe) file to update JD Edwards EnterpriseOne. Both files include the full software master with all of its directories. If you have a size restriction, such as when downloading from the internet, you can use the CAB file (which is always smaller than the executable file), but you will need to use a third-party application to uncompress the CAB file.

### 4.3.8 Confirming the Software Master

After you create the software master image, you should visually check the directory structure and XML database to verify that the image is complete.

To confirm the software master:

1. In Windows Explorer, open the image path directory.
2. Verify that the package directory includes these objects:
  - Install manager files.
  - Planner directory.
  - A data cab in the planner directory.
  - A directory that is named the same as the package.
  - Cab files in the package directory.
3. Verify that the \$TEMP\final directory includes these two files:
  - packagename.cab
  - packagename.exe
4. Open the XML database in the \$TEMP\master directory, and verify that this database is populated with tables.

---

**Note:** If the package includes new custom tables, verify that they are in the XML database and that they and their associated index change tables have records.

---

### 4.3.9 Moving the Software Master to a CD ROM

After the software master process has been completed and confirmed, the image can be burned onto a CD.

To move the software master to a CD ROM:

Using a CD burner and CD burner software, copy the packagename.cab or packagename.exe file from the \$TEMP/final directory on to the CD ROM.

### 4.3.10 Updating JD Edwards EnterpriseOne with the Software Master

This task explains how to update JD Edwards EnterpriseOne using the software master that you created. To perform this update, access the Work with Updates (P96470) application from the System Installation Tools (GH961) menu. This process includes setting merge flags, running the self-extracting executable (package\_name.exe), running software updates, and executing the installation workbench.

To update JD Edwards EnterpriseOne using a software master:

1. Run the package\_name.exe executable file from the CD.

This process installs the package into the Planner\Package directory on the deployment server, and extracts the package\_name.mdf and package\_name.ldf files into the Planner\Data directory. The process also creates the software master image in the PLANNER directory.

---

**Note:** You can also use the CAB file to create the software master image on the deployment server, but you need to use a third-party application to uncompress the file.

---

2. If InstallManager.exe exists on the CD, start the installation by double-clicking it.

If this file is not on the CD, double-click the package\_name.exe file to self-extract and uncompress the CAB file. After it is uncompressed, Install Manager will automatically launch.

3. On Installation Setup Screen, click Next.
4. On Installation Setup Type, verify that the machine has the necessary disk space and that the install path is correct, and then click Finish.

When the installation is complete, the Install Manager displays the "Installation Complete" window.

5. Click OK.
6. On the Deployment Server, sign in to the Planner environment (JDEPLAN).
7. Type **GH9612** in the Fast Path field.
8. Select Application Software Updates (P96470).
9. On Work with Software Updates, click Find.
10. Select the product package to install, and click Next.

11. On Software Update Environment Selection, select the path code on which to install the package, and click Next.

If you select unattended mode, the Installation Workbench will automatically perform all the workbench tasks to complete the installation. If you do not select unattended mode, the Installation Workbench tasks must be done manually.

If you select backup, the process will create a PAR file (PRJ\_package\_name\_pathcode\_60\_99.par) in the Planner\Package\Package\_Name\PathCode.bak directory. All objects that are affected by the update will be backed up in the PAR file.

The Installation Workbench should stop after the Table Conversion workbench process is complete.

12. Review the PDFs and the log files that were created by the table conversions, and then continue with Installation Workbench.

When Installation Workbench is complete, the system returns to the Work with Software Updates screen, and the selected software update should have the Completed Normally install status.

13. Click Close to quit Work with Software Updates.
14. Review all PDFs that were generated on the deployment server by the software update process, and verify that each of report is free of errors.

If a report has errors or does not appear to have run, review the jde.log file to determine the source of the problem. This table provides descriptions of the Software Update reports.

Report	Version	Description
R98405 - Application	XJDE0001	Table Conversion/Merge Driver for Application Tables: Master UBE for creating and regenerating application tables. This report should have only one record—Table and Index Creation (F984072). The status of the record should be, "Completed Normally."
R98407	XJDE0001	Table and Index Creation: Launched by R98405 UBE, which creates or regenerates all tables that are specified in the table change table. Page one of the report includes source, target, and environment information. Page two includes the overall status of the table creation in green or red text. Subsequent pages list the status of each table and index to be created or regenerated.
R98405 - Control	XJDE0001	Table Conversion/Merge Driver for Control Tables: The master UBE that specifies the control table merge UBEs to launch. It should list the Data Dictionary UBE (R989200P), and the User Defined Codes UBE (R9600042). The status on all should be, "Completed Normally."
R989200P	XJDE0001	Data Dictionary Merge: Launched by R98405. This UBE merges all data dictionary changes that are specified in the data dictionary change tables into the existing data dictionary. Page one of the report has source, target, and environment information. Subsequent pages list the status of each data dictionary addition or change. At the end of the individual listings, the overall status of the data dictionary merge is listed in green or red text. The final page has detailed summary information.

Report	Version	Description
R9600042	XJDE0001	User Defined Code Merge: Launched by R98405. This UBE merges all user-defined code changes that are specified in the UDC change tables into the existing UDC tables. Page one of the report lists source, target, and environment information. Subsequent pages list the status of each UDC addition or change. The final page has detailed summary information.
R9690002	XJDE0001	JD Edwards EnterpriseOne Solution Explorer Merge: Launched by R98405. This UBE merges all JD Edwards EnterpriseOne Solution Explorer changes that are specified in the SE change tables into the existing SE tables. Page one of the report lists source, target, and environment information. Subsequent pages list each addition, change, or deletion in the change tables. The final page has detailed summary information.
R96911002	XJDE0001	Favorites Merge: Launched by R98405. This UBE merges all Favorites changes that are specified in the Favorites change tables into the existing Favorites tables. Page one of the report lists the source, target, and environment information. Subsequent pages list the status of each Favorites addition or change. The final page has detailed summary information.
R96914002	XJDE0001	Report Director Templates Merge: Launched by R98405. This UBE merges all changes that are specified in the Report Director Templates change tables into the existing Report Director Templates tables. Page one of the report lists the source, target, and environment information. Subsequent pages list the status of each addition or change. The final page has detailed summary information.
R96915002	XJDE0001	Tips Merge: Launched by R98405. This UBE merges all Tips changes that are specified in the Tips change tables into the existing Tips tables. Page one of the report lists the source, target, and environment information. Subsequent pages list the status of each addition or change. The final page has detailed summary information.
R96951002	XJDE0001	Service Constant Merge: Launched by R98405. This UBE merges all service constant data that is specified in the change table into the existing Service Constant table. Page one of the report lists source, target, and environment information. Subsequent pages list the status of each service constant addition. This merge is only used for release 8.97.

Report	Version	Description
R98405 - Spec	XJDE0001	Table Conversion/Merge Driver for specification tables: Master UBE for specification table merge. This report should have only one record—Specification Merge (F98710). The status of this record should be, "Completed Normally."
R98700	ZJDE0002	Specification Merge: Launched by R98405. This UBE adds, replaces, and merges all specified object specifications into the selected Central Objects path code. This report is the standard specification merge report. Page one of the report lists the source, target, and environment information. Page two summarizes information about the merge status. The remaining pages include the individual object action and status.

15. On the Deployment Server, sign in to the DEP900 environment.
16. Find the package in Oracle's JD Edwards EnterpriseOne Package Assembly (P9601), and build and deploy the update package to a client workstation.

## 4.4 Deleting a Software Master

This section lists the form that is used to delete software masters. All records that are associated with the definition, such as the change table configurations and repair director records, will be deleted when you delete the software master.

### 4.4.1 Form Used to Delete Software Masters

Form Name	FormID	Navigation	Usage
Work With Software Mastering	W9640B	Product Packaging (GH962), Product Packages	Delete a software master.

## 4.5 Working with a Software Master Definition

This section provides overviews of software master revisions and software master validation and discusses how to:

- Revise a software master definition.
- Load a subdirectory from an enterprise network directory.
- Copy a software master definition.
- Proof a software master definition.

### 4.5.1 Understanding Software Master Revisions

After you have completed the definition of a software master, you can revise any of the information, including the build steps and directory structure. You also can copy the definition information to a new software master definition, delete the definition, and proof and validate it.

You can revise a software master definition that you have created. This revision enables you to modify the information that you entered into the JD Edwards EnterpriseOne CD Configuration Director, as well as the build steps and directory structure that is included with the template that you chose.

The system uses icons to indicate the status of a software master. The icons have these meanings:

Icon	Meaning
Gray Partial Cog	The status code is 10: In Definition.
Gray Full Cog	The status code is 20: Defined.
Gray Cog with Green Check Mark	The status code is 30: Validated.
Yellow-to-Gray Cyclic Arrows	The status code is 40: Processing.
Red Octagon	The status code is 50: Failed.
Gold Cog	The status code is 60: Built.

## 4.5.2 Understanding Software Master Validation

You run a report to validate the software master definition that you created. The report runs in final mode, and it changes the status of the build steps that do not have errors. All steps without errors whose status was In Definition or Defined are promoted to the status of Validated. The report lists each build step that will run when you create the master. The report lists whether the build step is validated. If it is validated, the build step will work properly when you are creating the software master. Otherwise, the build step will have either a warning or error message.

Warning messages will not stop the creation of a software master. The message specifies what the warning is. If a build step has a warning message, you should verify that the build step is as you want it before proceeding.

Error messages will stop the creation of a software master. You must fix any build steps that contain error messages. The message specifies the error.

## 4.5.3 Forms Used to Revise Software Masters

Form Name	FormID	Navigation	Usage
Work With Software Mastering	W9640B	Product Packaging (GH962), Product Packages	Select a software master to revise, proof, validate, or delete.
CD Revisions	W9640G	From Work With Software Mastering, select a software master to revise.	Verify and revise specifications for a software master. Run one or more steps.
Copy CD Configuration	W9640C	From Work with Software Mastering, select a software master and click Copy.	Create a new master based on an existing one.

## 4.5.4 Revising a Software Master Definition

Access the Work With Software Mastering form.



#### 4.5.4.1 CD Information

Select the CD Information tab.

##### **Status**

Displays the status of a build step or directory structure within the mastering process. For example, the build step or directory structure could be at a status of defined, validated, or built. Use the visual-assist button to view a complete list of values for this field.

##### **Comment**

Displays short informational comments that are provided by the system about each step of the mastering process. You can enter comments into this field, but the system will overwrite any text that is currently in the field during the next mastering process.

##### **Image Path**

Overfills the directory path where you want the software master to reside on the enterprise. The directory that you indicate will be the root directory for the creation of the final software master.

#### 4.5.4.2 Build Steps

Select the Build Steps tab.

##### **Sequence**

Displays the order in which the system processes build steps, usually in increments of 10 (for example, **10**, **20**, **30**, and so on).

Verify the order in which you want the step performed during the creation of the software master definition. You can use the same sequence number for multiple build steps.

##### **Type**

Displays the type of CD being mastered. Verify which component type you want the build step to perform. For example, you can define steps to create file structures, build a package, or create INF files.

##### **Status**

Verifies the current status of the build step, which can indicate whether to run the particular step during the mastering process. For example, if you need to perform a manual step during the mastering process, you can manually complete the step, change the step's status to **60** (Built) by choosing the step, and then, from the View menu, select Execute Step.

##### **Target Folder**

Displays the name of the target folder, such as **planner** or **updatepkg**. The system populates this field based upon the template that was used to define the software master. Change the name of the target folder only if you added the folder yourself.

##### **Executable**

Displays the software that might populate this field based upon the build step's definition in the software master template. To change this field, click the Executable button and use the Select a Windows Executable form to select an executable for this build step.

##### **Value/Parameter**

Verifies the value or parameter. To change this information, click the Value/Parameter button, and then enter a value or parameter that is appropriate to the step you chose.

This field is dynamic. The button name is either Value or Parameter, based upon the step that you select. The form that appears when you click the button also depends upon which step you select. For example, the Package Build step calls the Select Package Build form, from which you can select a package; and the Build Change Tables step calls the Work With Change Table Definitions form, from which you can modify the change table configuration.

#### 4.5.4.3 Directory Structure

Select the Directory Structure tab.

##### Source Path

Displays the directory on the enterprise that you want to use to populate the highlighted target folder. You can use an absolute (full) directory path name, or you can use a relative path name with a token. The system copies the data that is contained in the source path, depending upon how you filter those files (see the File Filter field description for this information), into the highlighted target folder of the software master.

Verifies the directory path of the source that will populate the component. This is the source path name of the data that you want to retrieve for the master and that you placed into the target folder. You can specify this path using either a relative or absolute path. See [Appendix A, "Troubleshooting Product Packaging"](#) for information about relative and absolute paths.

##### File Filter

Displays the filter, which you can designate, for the files that are contained in the source path. For example, you can use \*.\* to retrieve all files in the source path, or \*.exe to retrieve only executable programs.

Verify the file filter. The default is \*.\*. The file filter enables you to filter any data files that you retrieve from the source directory. For example, you can use \*.\* to retrieve all files in the source directory.

##### Full Path

Displays the names and functions. This field is dynamic and its name and function depend upon the build step that is highlighted. The possible names and functions are:

Value: The value that is needed to complete the highlighted build step. For example, if the build step is a batch process, the value field contains the name of the batch process and its version, such as **R95012\XJDE0001**.

Parameter: The parameters that are needed for the highlighted build steps that process an executable program.

The full (absolute) target path name of the highlighted directory, such as **Root/planner/updatepkg**.

#### 4.5.4.4 Task List

This table provides information for building the task list:

Task	Action
To run one or more steps:	<p>Select the build step that you want to run, and from the View menu, either select Execute Step to run just that step or select Run From Step to run that step plus any steps that follow it.</p> <p>This option runs the step and updates its status to 60 (Built). You might use this option for manual build steps. Manual steps are those that you need to perform yourself, such as an additional mastering step that is not part of the JD Edwards EnterpriseOne Product Packaging Tools. After you complete the manual step, use the Execute Step option to update the step to a status of 60 so that you can proceed to the next step in the process.</p>
To renumber the build steps:	<p>From the Form menu, select Renumber.</p> <p>The software renumbers the build steps, updating the Sequence field. The renumbering begins with 10 and increments each step by 10. The software keeps the build steps in the same sequence that you or the software master template established.</p>
To add a subdirectory:	<p>Click a directory for which you want to add a subdirectory and click Add. You can select any directory, including the Root directory.</p> <p>The software creates a new directory that is labeled Undefined and places it at the bottom of the structure.</p>

### 4.5.5 Loading a Subdirectory from an Enterprise Network Directory

Access the CD Revisions form.

1. Select the Directory Structure tab.
2. Select the directory that will be the root of the subdirectory that you want to load, which must already exist on the enterprise, and click Load.
 

This feature copies any enterprise directory structure into the software master definition.
3. On the Load Folders From Directory form, click Windows Folder.
4. On the Select Directory form, find and select a directory that you want to load into the software master, and then click OK.
5. On the Load Folders From Directory form, select one of these options:
  - Re-create subfolders
 

This option deletes the existing subdirectories and replaces them with the subdirectories that you want to load into the software master.
  - Append to existing
 

This option appends to the existing subdirectories the subdirectories that you want to load into the software master.
6. If necessary, select the Setup Source Directories check box.
 

During software master creation, by having this option enabled, the system automatically creates the structure of the source directories and copies all files within the source directories to the software master.

If you do not select Setup Source Directories, the software will create the source directory structure within the software master and the directories will be empty.

7. Click OK.

### 4.5.6 Copying a Software Master Definition

Access the Copy CD Configuration form.

You can copy a software master definition that you already created, which enables you to copy the information that you entered into the CD Configuration Director, the build steps, and the directory structure.

#### Type

The type of CD that is being mastered. Verify which component type you want the build step to perform. For example, you can define steps to create file structures, build a package, or create INF files.

In most cases, you will not need to change the default value. If you need to, however, you can change the type of the software master when making a copy. For example, you can change the type from a Setup CD to an ASU CD. This change does not change build steps or the directory structure. When you validate the software master, you will receive warning messages that the master does not match the default settings of the ASU CD template. This is because the master is actually based on the Setup CD template. You will not be prevented from creating the software master.

### 4.5.7 Proofing a Software Master Definition

Access the Work With Software Mastering form.

You can proof software master definitions. To do so, you create a report, which details each build step in the definition and lists the sequence of steps and processes that each step will run when creating the actual master.

Proofing a software master definition does not change any part of the status of a software master.

To proof a software master definition, perform one of these actions:

- To proof one software master definition, select the definition that you want to create, and then select Proof CD from the View menu.
- To proof all software master definitions for a specific release, select Proof Release from the View menu.

On the form that appears, enter the release number of the definitions that you want to proof.

## 4.6 Working with Software Master Templates

This section provides overviews of software master templates, the all build step processes, the ASU CD template, and the custom CD, and lists the forms used to work with software master templates.

### 4.6.1 Understanding Software Master Templates

The JD Edwards EnterpriseOne Product Packaging Tool uses software master templates. These templates provide default information when you are creating a new software master. For example, to distribute objects to a disconnected enterprise, you would build a master based on the ASU (Application Software Update) CD template.

Oracle's JD Edwards EnterpriseOne Mastering Director would use the ASU CD template to determine which director steps, build steps, and directory structure to use when creating the software master. You can configure the default information that is provided by the templates.

You use build steps to build a template. Build steps are the processes that are used to create a software master. You also can add, revise, copy, and delete software master templates. After you add a template, you can use the JD Edwards EnterpriseOne Mastering Director to create a software master definition using that template.

The processes for revising, copying, proofing, and deleting software master templates are identical to those for the software masters themselves.

## 4.6.2 Understanding the All Build Step Processes

A template can offer all of the following software mastering build step processes. Not all of these processes appear on every template.

### 4.6.2.1 Undefined

Type = 00

Performed Automatically = Not applicable

This build step is the default process when you are adding a new build step to a software master or template. After you add this build step, you need to define it for the master or template.

### 4.6.2.2 Create CD File Structure

Type = 01

Performed Automatically = Yes

This build step creates the physical directory structure at the image-path location that you specified when creating the software master definition.

### 4.6.2.3 Package Build

Type = 02

Performed Automatically = Yes

This build step builds the assigned package based upon the package assembly and build definitions.

### 4.6.2.4 Manual Build Step

Type = 03

Performed Automatically = No (See the specific template definition for complete information about the performance of this build step.)

This build step enables the administrator to instruct the user how to perform a process that has not yet been automated. You should verify the mastering steps and options of this step before beginning the master.

---

**Note:** The software mastering batch process stops when it encounters this build step. After you complete this build step, change its status to Succeeded (status code 60), and restart the software mastering batch process.

---

#### **4.6.2.5 Pre-Build Package**

Type = 04

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.6 Pre-Build Pristine**

Type = 05

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.7 Build Cross-Reference**

Type = 06

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.8 Build Data Dictionary**

Type = 07

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.9 Build In-House Package Database**

Type = 08

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.10 Build Planner Database**

Type = 09

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.11 Build Pristine Database**

Type = 10

Performed Automatically = Partial (See the specific template definition for complete information about the performance of this build step.)

This build step creates the pristine data tables in the final pristine database that corresponds to the given set of modules.

#### **4.6.2.12 Build Store & Forward Database**

Type = 11

Performed Automatically = Yes

For Oracle internal use only.

#### 4.6.2.13 Build Standalone Database

Type = 12

Performed Automatically = Yes

For Oracle internal use only.

#### 4.6.2.14 Build Change Tables

Type = 13

Performed Automatically = Yes

This build step builds the change tables based on the change table configuration that was set up as part of the software master definition.

#### 4.6.2.15 Master Central Objects

Type = 14

Performed Automatically = No

For Oracle internal use only.

#### 4.6.2.16 Compress

Type = 15

Performed Automatically = Partial (See the specific template definition for complete information about the performance of this build step.)

This build step compresses the final pristine JDE Masters database into the data.cab file in the final image path.

#### 4.6.2.17 Create INF

Type = 16

Performed Automatically = Yes

Do *not* modify this file manually. This build step updates the deployment.inf file that the installation applications use during the update installation.

#### 4.6.2.18 Copy CD Directory Files

Type = 17

Performed Automatically = Yes

This build step copies the built package to the target image path. This step also copies any additional files that are defined in the software master directory structure to the target image path.

---

**Note:** If you have already built the package and do not want the mastering process to build the package again, set the status of the Package Build step to 60. When you create the software master, the process bypasses the package build and, when it runs this Copy CD Directory Files step, copies the package.

---

#### 4.6.2.19 Virus Check

Type = 18

Performed Automatically = Partial

For Oracle internal use only.

#### **4.6.2.20 Windows Executable**

Type = 19

Performed Automatically = Partial (See the specific template definition for complete information about the performance of this build step.)

This build step enables you to specify a Microsoft Windows executable or batch file as part of the mastering process.

#### **4.6.2.21 Data Cleanup**

Type = 20

Performed Automatically = No

For Oracle internal use only.

#### **4.6.2.22 Build Package Database**

Type = 21

Performed Automatically = Yes

For Oracle internal use only.

#### **4.6.2.23 Batch Process**

Type = 22

Performed Automatically = Yes

This build step enables you to define a batch process to run as part of the software master definition.

### **4.6.3 Understanding the ASU CD Template**

The ASU CD is the main template that is used with the JD Edwards EnterpriseOne Product Packaging Tools. This template is the basis for delivering objects to distributed enterprises. The Product Packaging ASU Template data is delivered in the ASUTMPL900 configuration package on the Update Center. Use Change Assistant to download and deploy this package. This table shows the build steps and their sequence for an ASU CD:

<b>Sequence Number</b>	<b>Build Step (Type)</b>
10	Check Mastering Items (03 - Manual Build Step)
20	Cleanup Temporary Files (22 - Batch Process)
30	Create CD File Structure (01 - Create the CD File Structure)
40	Build (02 - Package Build)
60	Build Change Tables (13 - Build Change Tables)
70	Create/Load ASU Change Tables (22 - Batch Process)
80	Create/Load ASU Control Tables (22 - Batch Process)
90	Create/Load ASU Misc Tables (22 - Batch Process)



Sequence Number	Build Step (Type)
100	Create/Load New and Regen Tables (22 - Batch Process)
110	Create/Load ASU OL & VL Tables (22 - Batch Process)
120	ASU v/s Package Validation (22 - Batch Process)
130	Compress/Copy Database (15 - Compress)
140	Copy Package and Feature INFs (22 - Batch Process)
180	Copy CD Directory Files
190	Create INF
200	Create self-extracting executable
210	Validate the mastered CD

This table provides additional information pertaining to a particular build step within the ASU CD template:

Sequence Number	Build Step (Type) with Additional Information
10	Check Mastering Items (03 - Manual Build Step) Verify the mastering steps and options of the step before beginning the master. This build step must be executed manually before the Master CD.
20	Cleanup Temporary Files (22 - Batch Process) Launches the R967002 batch process using version XJDE0001. This process deletes the files and folders from the directory that you specify (for example, \$TEMP\data).
30	Create CD File Structure (01 - Create CD Structure) Creates the CD file structure on the Image path.
40	Build (02 - Package Build) Builds the defined update package. This build step can be set to Built status if the package has run through the Package Build application.
60	Build Change Tables (13 - Build Change Tables) Builds the change tables based on the configuration that was set up during the director phase.
70	Create/Load ASU Change Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0003. This process creates change tables in the XML database, populating the tables from the Change Table data source.
80	Create/Load ASU Control Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0001. This process creates control tables in the XML database, and populates the tables from the Control Table data source.
90	Create/Load ASU Misc Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0002. This process creates miscellaneous tables in the XML database, and populates the tables based on the selections that you made for the objects in the ASU package, the change tables, and the software master definition.

Sequence Number	Build Step (Type) with Additional Information
100	<p>Create/Load New &amp; Regen Tables (22 - Batch Process)</p> <p>Launches the R96700 batch process using version XJDE0001. This process creates tables in the XML database that are marked as new or regenerate in the table change table or the table conversion scheduler.</p>
110	<p>Create/Load ASU OL &amp; VL tables (22 - Batch Process)</p> <p>Launches the R9600400C batch process using version XJDE0001. This process copies to the XML database the Object Librarian, Versions List, and package assembly information for the master. Verify that the data source, which you can set in the processing options for this version, is set correctly.</p>
120	<p>ASU v/s Package Validation (22 - Batch Process)</p> <p>Launches the R9671 batch process using version XJDE0001. This process cross- validates information, ensuring that objects, tables, and specifications in the ASU package definition match with the object change table, table change table, index change table, and table conversion scheduler in the software master.</p>
130	<p>Compress/Copy Database (22 - Batch Process)</p> <p>Launches R9640E using version XJDE0003. This process compresses the ASU database XML files and copies them from the \$TEMP/data directory to the final software master image path as Data.cab.</p>
140	<p>Copy Package and Feature INFs (22 - Batch Process)</p> <p>Launches the R9600400E batch process using version XJDE0001. This process copies deployment features INFs, the package INFs, and the feature INFs from the deployment server to the \$TEMP directories.</p>
180	<p>Copy CD Directory Files (17 - Copy CD Directory Files)</p> <p>Copies additional specified items into the image path. For example, all install programs are copied to the root of the image path.</p>
190	<p>Create INF (16 - Create INF)</p> <p>Creates the deployment INF file into the image path directory.</p>
200	<p>Create self-extracting exe (22 - Batch Process)</p> <p>Launches the R9640J batch process using version XJDE0001. This process takes the entire software master image and creates a CAB file and a self-extracting executable file in the \$TEMP/final directory. You can use either of these files to deliver the software master. Uncompress the CAB file or run the executable file on the deployment server, and then refer to the Application Software Update Guide to update JD Edwards EnterpriseOne with the software master.</p>
210	<p>Validate the Mastered CD (22 - Batch Process)</p> <p>Launches R9640D version XJDE0001, which performs post-mastering validation on the master that has been created. Refer to the report output for details.</p>

## 4.7 Working with Custom CD Templates

The Custom CD is for general use; this CD type has no template. A JD Edwards EnterpriseOne administrator uses this CD type primarily to create backups of source code, documents, and so forth, while still using the JD Edwards EnterpriseOne Product Packaging Tools. This CD type enables you to enter a directory structure that contains all of the files that you want loaded on to a custom CD image. After the process is completed, the administrator can move that image to a CD burner to create the actual CD.

### 4.7.1 Forms Used to Work with Software Master Templates

Form Name	FormID	Navigation	Usage
Work With CD Templates	W9640B	Advanced Operations menu (GH9622), Mastering Templates	Select a template to revise, proof, validate, or delete.
Template Director	W9640D	On the Work With CD Templates form, click Add.	Add a new template
CD Revisions	W9640G	On the Work With CD Templates form, select a template.	Revise a template.
Copy CD Configuration	W9640C	From Work with Software Mastering, select a template and click Copy.	Create a new template based on an existing one.

**See Also:**

- [Revising a Software Master Definition.](#)
- [Loading a Subdirectory from an Enterprise Network Directory.](#)
- [Copying a Software Master Definition.](#)
- [Proofing a Software Master Definition.](#)



---

## Working with Change Table Configurations

---

This chapter contains the following topics:

- [Section 5.1, "Understanding Change Table Revisions"](#)
- [Section 5.2, "Working with Change Table Configurations"](#)
- [Section 5.3, "Working with Change Table Director"](#)

### 5.1 Understanding Change Table Revisions

If you need to revise or add change tables, you can build and submit change table configurations to a software master after you have created the software master. Disregard this task if you have not yet created the software master. If the template that you select for the software master requires change tables, the software master will build the change tables that you specify during the setup of the software master definition.

### 5.2 Working with Change Table Configurations

You copy change table configurations from one master definition to another master definition. This process copies all change table configurations associated with the master from which you are copying.

#### 5.2.1 Forms Used to Work with Change Tables

Form Name	FormID	Navigation	Usage
Work With Change Table Definitions	W9642Z	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Change Table Configurations (P9642)	Used to revise or delete change tables.

Form Name	FormID	Navigation	Usage
Copy Change Table Definitions	W9642A	In the Work With Change Table Definitions form, select a CD, then select one of its change table definitions and click Copy.	Used to create a new table definition based on an existing one.

## 5.2.2 Setting Up Change Table Definitions

Access the Work With Change Table Definitions form.

**Figure 5–1** *Change Table Configurations [Work with Change Table Definitions] form*

Change Table Configurations - [Work With Change Table Definitions]

File Edit Preferences Form Row Window Help

Select Find Add Copy Del... Close Seg... New... Dis... Ago Links Row OLE ... Internet

CD Name  Release

Description

The Configuration Status must be Validated before the associated Change Table Batch Application can be submitted. The checked items show those that are ready to run or have already run successfully.

	Source Release	Target Release	Source Environment	Change Table Type	Batch Application	Version	Configuration Status

**Source Release**

The system release number under which the source pristine tables were built.

**Target Release**

The system release number for which you are building the software master.

**Source Environment**

The name of the source environment where the pristine control tables reside.

**Change Table Type**

A user defined name or remark.

**Batch Application**

The number that identifies the batch or interactive program (batch or interactive object).

**Version**

A user-defined set of specifications that control how applications and reports run. You use versions to group and save a set of user-defined processing option values and data selection and sequencing options. Interactive versions are associated with applications (usually as a menu selection). Batch versions are associated with batch jobs or reports. To run a batch process, you must choose a version.

**Configuration Status**

A user defined name or remark.

## 5.3 Working with Change Table Director

If the template that you select for the software master requires change tables, the software master will build the change tables that you specify during the setup of the software master definition.

### 5.3.1 Forms Used to Work With Change Table Director

Form Name	FormID	Navigation	Usage
Work With Change Table Definitions	W9642Z	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Change Table Configurations (P9642)	Used to revise or delete change tables.
Change Table Director	W9642E	In the Work With Change Table Definitions form, click Add.	Guides you through the steps necessary in setting up a configuration for running the Change Table batch applications across multiple releases.
Target Release Selection	W9642F	Complete the fields and click Next.	Used to identify the name of the CD and then change the Target release necessary.
Source Environment Selection	W9642G	Complete the fields and click Next.	Used to select the name of the environment where the pristine control tables exit.



Form Name	FormID	Navigation	Usage
Batch Application Selection	W9642AA	Select the batch applications and click Next.	Used to select the Change Table Types that you wish to have updated.
Additional Change Table Definitions	W9642L	Select Quit to close the JD Edwards EnterpriseOne Change Table Director or select Continue to display the Target Release Selection form.	Used to either add additional change table definitions or to exit the JD Edwards EnterpriseOne Change Table Director.

### 5.3.2 Adding a Change Table Definition

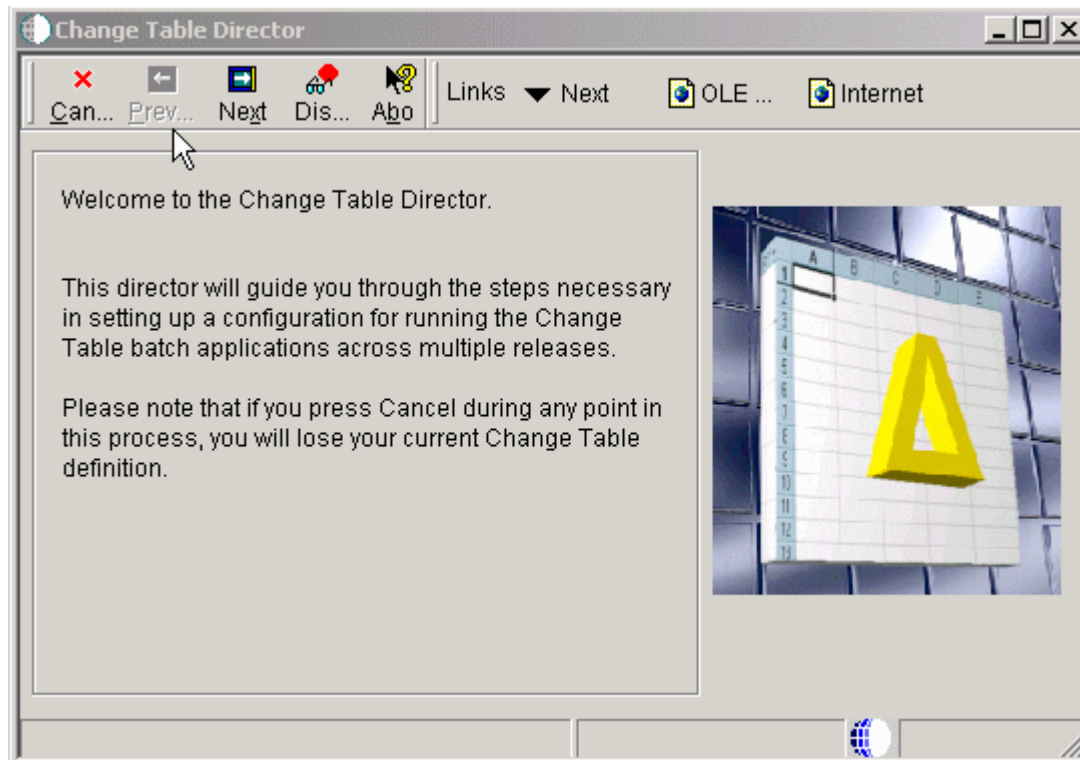
The software builds the change tables that you select and includes them with the software master.

#### 5.3.2.1 Change Table Director

Oracle's JD Edwards EnterpriseOne Change Table Director guides you through the steps necessary in setting up a configuration for running the Change Table batch applications across multiple releases.

Access the JD Edwards EnterpriseOne Change Table Director form.

**Figure 5–2** Change Table Director form



### 5.3.2.2 Target Release Selection

Each Change Table configuration must be associated with a previously defined CD. Identify the name of the CD and then change the Target release necessary. In most cases, the target release will be the same as the defined release of the CD.

Access the Target Release Selection form.

**Figure 5–3** Change Table Configurations [Target Release Selection] form

Change Table Configurations - [Target Release Selection]

File Edit Preferences Form Window Help

Can... Next Dis... Abo Links Previo... OLE ... Internet

Each Change Table configuration must be associated with a previously defined CD. Identify the name of the CD and then change the Target release if necessary. In most cases, the target release will be the same as the defined release of the CD.


CD Name

Description

Release  A7.1 Base

Target Release  E812 Release 8.12

The Target Release cannot be the same as the Source Release. Use the director to add or select a unique release value.



**CD Name**

The unique name given to the software master.

**Description**

A short description of the software master

**Release**

The release number defined in the software master.

**Target Release**

The system release number for which you are building the release master.

**5.3.2.3 Source Environment Selection**

The Source Environment Selection is used to select the name of the environment where the pristine control tables exist. This information is used along with the current environment to build the Change Table records.

Access the Source Environment Selection form.

**Figure 5–4** *Change Table Configurations [Source Environment Selection] form*

Change Table Configurations - [Source Environment Selection]

File Edit Preferences Form Window Help

Can... Prev... Next Dis... Algo Links Previo... OLE ... Internet

Choose the name of the environment where the pristine control tables exist. This information is used along with the current environment to build the Change Table records.

To see object mapping information for your selected environment, you may choose to take the Object Configuration Manager menu exit.

Source Environment

Description

Source Release

**Source Environment**

The name of the source environment where the pristine control tables reside.

**Description**

A user defined name or remark.

### Source Release

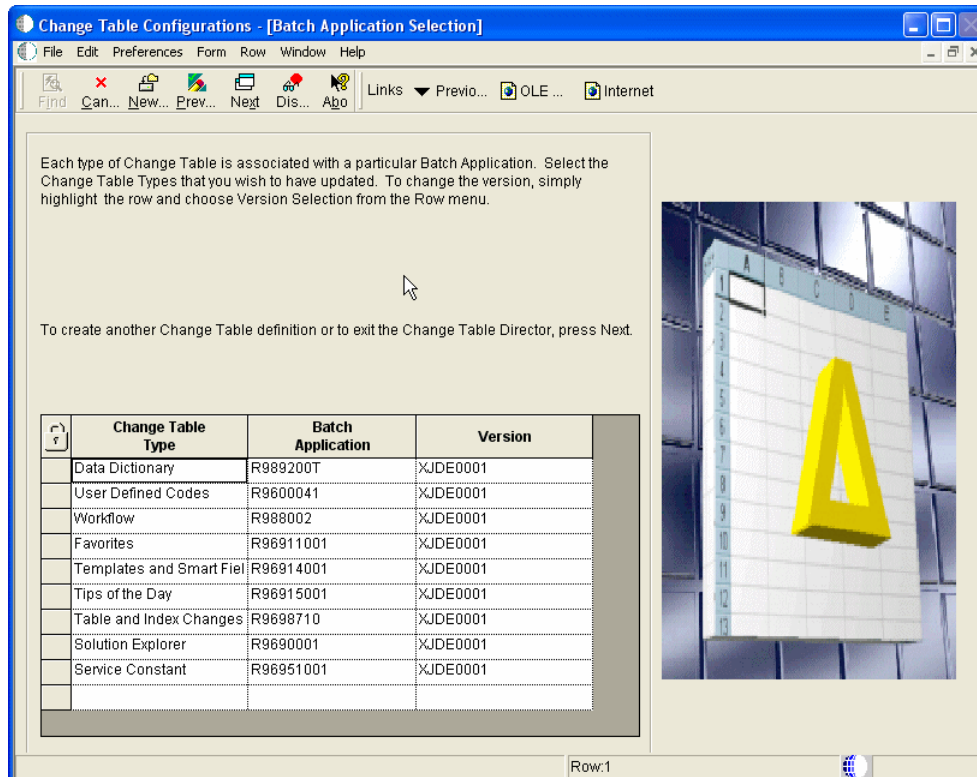
The release number defined in the release master.

#### 5.3.2.4 Batch Application Selection

Each type of Change Table is associated with a particular Batch Application. Select the Change Table Types that you wish to have updated. To change the version, simply highlight the row and choose the Version Selection form the Row menu.

Access the Batch Application Selection form.

**Figure 5–5** *Change Table Configurations [Batch Application Selection] form*



**Change Table Type**

A user defined name or remark.

**Batch Application**

Additional text that further describes or clarifies a field in the system.

**Version**

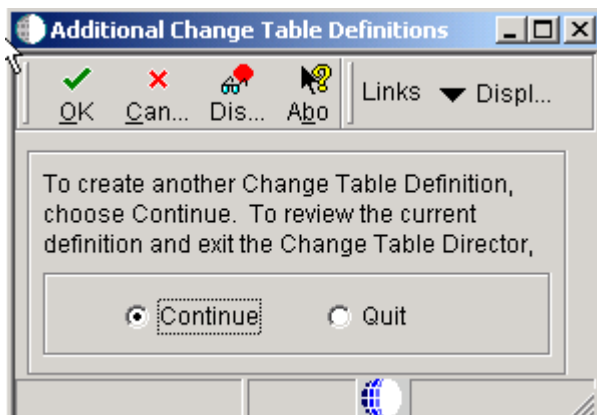
A user-defined set of specifications that control how applications and reports run. You use versions to group and save a set of user-defined processing option values and data selection and sequencing options. Interactive versions are associated with applications (usually as a menu selection. Batch versions are associated with batch jobs or reports. To run a batch process, you must choose a version.

**5.3.2.5 Additional Change Table Definitions**

The Additional Change Table Definitions form is used to either add additional change table definitions or to exit the Change Table Director.

Access the Additional Change Table Definitions form.

**Figure 5–6** Additional Change Table Definitions form

**Continue**

Select Continue to display the Target Release Selection form where you can then repeat all of the steps for a different source and target combination.

**Quit**

Select Quit to close the JD Edwards EnterpriseOne Change Table Director.

---

## Working with the Data Cleanup Procedure Application

This chapter contains the following topics:

- [Section 6.1, "Understanding the Data Cleanup Procedure Application"](#)
- [Section 6.5, "Executing a Data Cleanup Procedure and Build Step"](#)
- [Section 6.3, "Adding a Data Cleanup Procedure"](#)
- [Section 6.4, "Copying a Data Cleanup Procedure"](#)
- [Section 6.5, "Executing a Data Cleanup Procedure and Build Step"](#)

### 6.1 Understanding the Data Cleanup Procedure Application

JD Edwards EnterpriseOne Product Packaging Tool includes an application that enables you to add, revise, copy, and delete a set of data cleanup steps. You define how you want these steps to clean up the software master data.

### 6.2 Working with the Data Cleanup Procedure Application

You can revise the information and steps within a data cleanup procedure that you have already created. If you revise a data cleanup procedure, those revisions are reflected wherever that procedure is used, such as when the revised procedure is nested within another procedure.

#### 6.2.1 Forms Used to Work with Data Cleanup Procedures

Form Name	FormID	Navigation	Usage
Work with Procedure	W9646A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Data Cleanup Procedure Templates	Used to execute or delete a data cleanup procedure.

## 6.2.2 Using the Data Cleanup Procedure Application

Although this JD Edwards EnterpriseOne Product Packaging Tool function is primarily used to create data cleanup procedures, you can use it to create any type of procedure to run a batch process or Microsoft Windows executable.

Access the Work With Procedure form.

**Figure 6–1** *Work with Procedure form*

Release	Name	Description	Status	Type	Comment	User ID

### **Release**

The system release to which the data cleanup applies.

### **Name**

The unique name given to a data-cleanup procedure.

### **Description**

A short description of the data-cleanup procedure.

### **Status**

The status of the data-cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. For example, the procedure could be at a status of defined, validated, or built. Before executing the data cleanup procedure (whether from the software master or from the data-cleanup procedure application), verify that the status of the procedure, and all of its build steps that you want to run with it, are validated.

### **Type**

The type of procedure being run.

### **Comment**

A short informational comment that you provide about the data-cleanup procedure.



**User ID**

The code that identifies a user profile.

## 6.3 Adding a Data Cleanup Procedure

This function is used to create various data cleanup procedures to run a batch process.

### 6.3.1 Forms Used to Add a Data Cleanup Procedure

Form Name	FormID	Navigation	Usage
Work With Procedure	W9646A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Data Cleanup Procedure Application (P9646)	Used to create various procedures to run a batch process.
Procedure Definition Director	W9646B	In the Work With Procedure form, click Add.	Enables you to add procedure information and the steps that are necessary to clean up data.
Procedure Revision Information	W9646D	On the Work with Procedure form, select a procedure.	Revise or execute a data cleanup procedure.

### 6.3.2 Adding Data Cleanup Procedures

This enables you to add procedure information and the steps that are necessary to clean up data.

#### 6.3.2.1 Procedure Definition Director

Oracle's JD Edwards EnterpriseOne Procedure Definition Director will take you through the steps necessary to define a procedure.

Access the Procedure Definition Director form.

#### 6.3.2.2 Add Procedure

The Add Procedure form is used to enter information about the new procedure.

Access the Add Procedure form.

**Figure 6–2 Data Cleanup Procedure Application [Add Procedure] form**

Data Cleanup Procedure Application - [Add Procedure]

File Edit Preferences Form Window Help

Can... Prev Next Dis... [Help Icon]

Links ▼ Prev OLE ... Internet

Enter information about your new procedure.

Enter the unique name and other information related to the procedure

Name

Description

Comment

Status  In Definition

Procedure Release

Template Name

**Name**

The unique name given to a data-cleanup procedure.

**Description**

A short description of the data cleanup procedure.

**Comment**

A short informational comment that you provide about the data cleanup procedure.

**Status**

The status of the data cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. Before executing the data-cleanup procedure (whether from the software master or from the data-cleanup procedure application), verify that the status of the procedure, and all of its build steps that you want to run with it, are validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Procedure Release**

The system release number to which the data cleanup procedure applies.

**Template Name**

The unique name given to a data cleanup procedure template.

**6.3.2.3 Procedure Revision Information**

These options execute the entire procedure. If the procedure is executed successfully, the software updates the procedure and the build step statuses to 60 (Built).

Access the Procedure Revision form.

**Figure 6–3** Data Cleanup Procedure Templates [Procedure Revision] form

The screenshot shows a software window titled "Data Cleanup Procedure Application - [Procedure Revision]". The window has a menu bar with "File", "Edit", "Preferences", "Form", "View", "Window", and "Help". Below the menu bar is a toolbar with buttons: "OK", "Can...", "Add", "Del...", "Ren...", "Refr...", "Dis...", "Ago", "Links", "Run M...", "OLE ...", and "Internet". The main area has two tabs: "Information" (selected) and "Steps". The "Information" tab contains the following fields:

- Name:** A text box containing the word "Procedure".
- Description:** A text box containing the word "Procedure".
- Status:** A dropdown menu showing "10" and a button labeled "In Definition".
- Comment:** An empty text box.
- Release:** A text box containing "E812".

On the right side of the form, there is a graphic of four interlocking yellow gears on a cloudy background.

**Name**

The unique name given to a data-cleanup procedure.

**Description**

A short description of the data cleanup procedure.

**Status**

The status of the data cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. Before executing the data-cleanup procedure (whether from the software master or from the data-cleanup procedure application), verify that the status of the procedure, and all of its build steps that you want to run with it, are validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Comment**

A short informational comment that you provide about the data cleanup procedure.

**Release**

The CD release number to which the data-cleanup applies.

**6.3.2.4 Procedure Revision Steps**

This option executes the data cleanup procedure step. If the step is executed successfully, the software updates the step status to 60 (Built). If you execute a data-cleanup procedure step, all of its subordinate steps are also executed, including any sub-procedures.

Access the Procedure Revision form.

**Figure 6–4 Data Cleanup Procedures Templates [Procedure Revision] form**

The screenshot shows a software application window titled "Data Cleanup Procedure Application - [Procedure Revision]". The window has a menu bar with "File", "Edit", "Preferences", "Form", "View", "Window", and "Help". Below the menu bar is a toolbar with buttons for "OK", "Can...", "Add", "Del...", "Ren...", "Refr...", "Dis...", and "Ago". There are also icons for "Links", "View", "OLE...", and "Internet".

The main area of the window is divided into two tabs: "Information" and "Steps". The "Information" tab is active and contains the following fields:

- Status:** A text box containing "30" and a dropdown menu showing "Validated".
- Type:** A text box containing "01" and a dropdown menu showing "Generic UBE".
- Sequence:** A text box containing "10".
- Description:** A text box containing "Step 1".
- Comment:** A text box containing "Step 1".
- Object Name:** An empty text box.
- Version:** An empty text box.

The "Steps" tab is currently empty.

## 6.4 Copying a Data Cleanup Procedure

The JD Edwards EnterpriseOne Product Packaging Tool function is primarily used to create data cleanup procedures.

## 6.4.1 Forms Used to Copy a Data Cleanup Procedure

Form Name	FormID	Navigation	Usage
Work With Procedure	W9646A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Data Cleanup Procedure Application (P9646)	Used it to create various procedures to run a batch process.
Copy Procedure	W9646L	In the Work with Procedure form, select a procedure and click Copy.	Used to create a new data cleanup procedure based on an existing one.

## 6.4.2 Copying a Data Cleanup Procedure

Access the Procedure Definition Director form.

**Figure 6–5** Copy Procedure form



### Copy From Procedure

The unique name given to a data cleanup procedure.

### Copy To Procedure

The unique name given to a data cleanup procedure.

## 6.5 Executing a Data Cleanup Procedure and Build Step

You can execute a data cleanup procedure and build step from the data cleanup application.

Although this task explains how to execute procedures from the data cleanup application, you can also use the software master build steps to define a step that executes the data cleanup.

## 6.5.1 Forms Used to Execute a Data Cleanup Procedure and Build Step

Form Name	FormID	Navigation	Usage
Work With Procedure	W9646A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Data Cleanup Procedure Application (P9646)	Use to create various procedures to run a batch process.
Procedure Revision	W9646D	In the Work with Procedure form, select a procedure.	Used to revise or execute a data cleanup procedure.

## 6.5.2 Executing a Data Cleanup Procedure and Build Step

Access the Procedure Revision form.

### Name

The system release number to which the data cleanup applies.

### Description

The unique name given to the data cleanup procedure.

### Status

The status of the data cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. Before executing the data-cleanup procedure (whether from the software master or from the data-cleanup procedure application), verify that the status of the procedure, and all of its build steps that you want to run with it, are validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

### Comment

A short informational comment that you provide about the data cleanup procedure.

### Release

The CD release number to which the data-cleanup applies.



---

## Working with Advanced Operations

---

This chapter contains the following topics:

- [Section 7.1, "Working with CD Templates"](#)
- [Section 7.2, "Converting Relative and Absolute Paths"](#)
- [Section 7.3, "Working with Data Cleanup Procedure Templates"](#)
- [Section 7.4, "Viewing a Table Change Table"](#)
- [Section 7.5, "Viewing the Index Change Tables"](#)
- [Section 7.6, "Viewing the Control Change Tables"](#)

### 7.1 Working with CD Templates

This section provides an overview of CD templates and discusses how to:

- Work with CD Templates.
- Add a New Template

#### 7.1.1 Understanding CD Templates

A JD Edwards EnterpriseOne administrator uses this CD template primarily to create backups of source code, documents, and so forth, while still using JD Edwards EnterpriseOne Product Packaging Tools.

#### 7.1.2 Forms Used with Mastering Templates

Form Name	FormID	Navigation	Usage
Work With CD Templates	W9640B	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Advanced Operations, Mastering Templates (P9640)	Used to select a template to revise, proof, validate, or delete.
CD Revisions	W9640G	In the Work With CD Templates form, select a template.	Used to revise a template.

Form Name	FormID	Navigation	Usage
Template Director	W9640D	In the Work With CD Templates form, click Add.	Used to add a new template
Template Information	W9640H	Complete the fields and click Next.	Used to provide the release level, CD type and a short description. The template name will be generated automatically.
CD Revisions	W9640G	Complete the fields and click Next.	Lets you set up your CD information, build steps, directory structure, and director steps.
Copy CD Configuration	W9640C	In the Work With Software Mastering, select a template and click Copy.	Used to create a new template based on an existing one.

### 7.1.3 Setting Up CD Templates

The CD template enables you to enter a directory structure that contains all of the files that you want loaded on to a custom CD image. After the process is completed, the administrator can move that image to a CD burner to create the actual CD.

Access the Work With CD Templates form.

**Figure 7–1 Product Packages [Work with CD Templates] form**

Release	Build Phase	Build Phase Description	CD Type	Type Description	CD Name	Description
---------	-------------	-------------------------	---------	------------------	---------	-------------

**Release**

The system release number of the software master. use the Search button to review a complete list of values for this field.

**Build Phase**

This is the phase of development that your software master represents, such as an alpha, a beta, or a general availability phase. You can add or change the phases, since this is a user defined code. Use the visual-assist button to view a complete list of values for this field.

**Build Phase Description**

A user defined name or remark.

**CD Type**

Type of CD being mastered.

**Type Description**

A user defined name or remark.

**CD Name**

The unique name given to a software master.

**Description**

A short description of the software master.

**Comment**

A short informational comment provided by the system about each step of the mastering process. You can enter your own comments into this field, but the system will overwrite any text currently in the field during the next mastering process.

**Definitions Date**

The date the CD was defined.

**One Off CD**

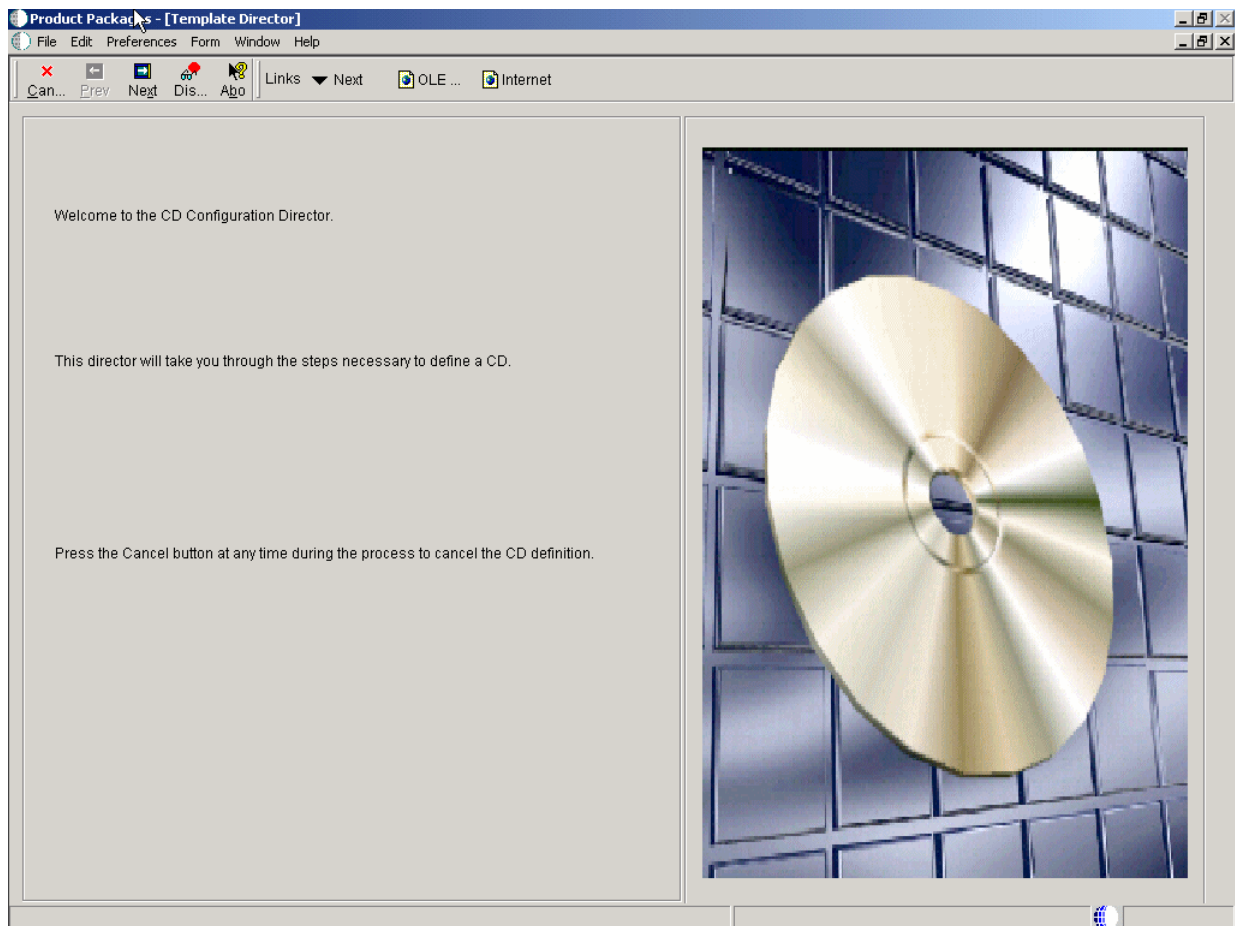
Future use.

## 7.1.4 Adding a New CD Template

Access the Template Director form.

### 7.1.4.1 Template Director

Oracle's JD Edwards EnterpriseOne Template Director is used to guide you through the steps necessary to define a CD.

**Figure 7-2** *Product Packages [Template Director] form*

#### 7.1.4.2 Template Information

The Template Information form is used to provide the release level, CD type and a short description. The template name will be generated automatically.

Access the Template Information form.

**Figure 7–3** *Product Packages [Template Information] form*

**Release**

The system release number of the software master. Use the Search button to review a complete list of values for this field.

**Type**

The type of procedure being run.

**Build Phase**

This is the phase of development that your software master represents, such as an alpha, a beta, or a general availability phase. You can add or change the phases, since this is a user defined code. Use the visual-assist button to view a complete list of values for this field.

**Description**

A user defined name or remark.

**7.1.4.3 CD Revisions**

The CD Revisions form lets you set up your CD information, build steps, directory structure, and director steps.

Access the CD Revisions form.

**Figure 7-4 Product Packages [CD Revisions] form**

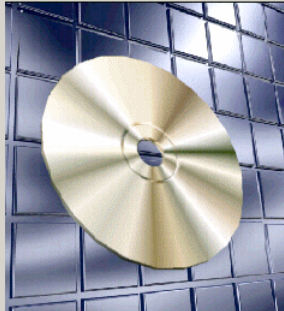
**Product Packages - [CD Revisions]**

File Edit Preferences Form View Window Help

OK Can... Prev... Pro... Vie... Set... Add... Del... Loa... Dis... Links Previo... OLE ... Internet

**CD Information** Build Steps Directory Structure Director Steps

Name	<input type="text" value="E812-09"/>	
Type	<input type="text" value="09"/>	Application Software Update
Release	<input type="text" value="E812"/>	Release 8.12
Build Phase	<input type="text" value="03"/>	General Availability
Status	<input type="text" value="10"/>	In Definition
Description	<input type="text" value="Template for E812-Application Software Update"/>	
Comment	<input type="text"/>	



1. Enter your CD Information.

- **Status**



Displays the status of a build step or directory structure within the mastering process. For example, the build step or directory structure could be at a status of defined, validated, or built. Use the visual-assist button to view a complete list of values for this field.

- **Comment**

Displays short informational comments that are provided by the system about each step of the mastering process. You can enter comments into this field, but the system will overwrite any text that is currently in the field during the next mastering process.

- **Image Path**

Overfills the directory path where you want the software master to reside on the enterprise. The directory that you indicate will be the root directory for the creation of the final software master.

## 2. Define your Build Steps.

- **Sequence**

Displays the order in which the system processes build steps, usually in increments of 10 (for example, **10**, **20**, **30**, and so on).

Verify the order in which you want the step performed during the creation of the software master definition. You can use the same sequence number for multiple build steps.

- **Type**

Displays the type of CD being mastered. Verify which component type you want the build step to perform. For example, you can define steps to create file structures, build a package, or create INF files.

- **Status**

Verifies the current status of the build step, which can indicate whether to run the particular step during the mastering process. For example, if you need to perform a manual step during the mastering process, you can manually complete the step, change the step's status to **60** (Built) by choosing the step, and then, from the View menu, select Execute Step.

- **Target Folder**

Displays the name of the target folder, such as **planner** or **updatepkg**. The system populates this field based upon the template that was used to define the software master. Change the name of the target folder only if you added the folder yourself.

- **Executable**

Displays the software that might populate this field based upon the build step's definition in the software master template. To change this field, click the **Executable** button and use the Select a Windows Executable form to select an executable for this build step.

- **Value/Parameter**

Verifies the value or parameter. To change this information, click the **Value/Parameter** button, and then enter a value or parameter that is appropriate to the step you chose. This field is dynamic. The button name is either Value or Parameter, based upon the step that you select. The form that appears when you click the button also depends upon which step you select.

For example, the Package Build step calls the Select Package Build form, from which you can select a package; and the Build Change Tables step calls the Work With Change Table Definitions form, from which you can modify the change table configuration.

3. Set up your Directory Structure.

- **Source Path**

Displays the directory on the enterprise that you want to use to populate the highlighted target folder. You can use an absolute (full) directory path name, or you can use a relative path name with a token. The system copies the data that is contained in the source path, depending upon how you filter those files (see the **File Filter** field description for this information), into the highlighted target folder of the software master.

Verifies the directory path of the source that will populate the component. This is the source path name of the data that you want to retrieve for the master and that you placed into the target folder. You can specify this path using either a relative or absolute path. See [Appendix A, "Troubleshooting Product Packaging"](#) for information about relative and absolute paths.

- **File Filter**

Displays the filter, which you can designate, for the files that are contained in the source path. For example, you can use \*.\* to retrieve all files in the source path, or \*.exe to retrieve only executable programs.

Verify the file filter. The default is \*.\*. The file filter enables you to filter any data files that you retrieve from the source directory. For example, you can use \*.\* to retrieve all files in the source directory.

- **Full Path**

Displays the names and functions. This field is dynamic and its name and function depend upon the build step that is highlighted. The possible names and functions are:

**Value:**The value that is needed to complete the highlighted build step. For example, if the build step is a batch process, the value field contains the name of the batch process and its version, such as **R95012\XJDE0001**.

**Parameter:**The parameters that are needed for the highlighted build steps that process an executable program.

The full (absolute) target path name of the highlighted directory, such as **Root/planner/updatepkg**.

4. Define your Director Steps.

- **Package/Change Table Form**

Indicates the package form and change table form used.

- **Software Update Form**

Future use.

## 7.1.5 Copying CD Configuration

You can copy a software master definition that you have already created. This copies the information that you entered into the JD Edwards EnterpriseOne CD Configuration Director, the build steps, and the directory structure.

Access the Copy CD Configuration form.

**Figure 7–5** *Product Packages [Copy CD Configuration] form*

**Product Packages - [Copy CD Configuration]**

File Edit Preferences Window Help

OK Cancel Disc Abort Links Display OLE Internet

**Copy From**

Name: E812-09

**Copy To**

Name: -09 (Name automatically generated from Release and Type)

Description: Template for -Application Software Update

Status: 10 In Definition

Release: A7.1 Base

Type: 09 Application Software Update

**Copy From**  
The Copy From fields:

Name is the unique name given to a software master.

### **Copy To**

The Copy To fields:

Name is the unique name given to a software master and is automatically generated from release and type.

Description is a short description of the software master.

Status displays the status of a build step or directory structure within the mastering process.

Release is the system release number of the software master.

Type is the type of CD being mastered.

## **7.2 Converting Relative and Absolute Paths**

This section provides an overview of relative and absolute paths and discusses how to:

- Use relative path tokens
- Convert relative and absolute paths.

### **7.2.1 Understanding Path Tokens**

You use relative path tokens within a field. The software enables you to convert relative path tokens and absolute path names for those fields that accept them within the JD Edwards EnterpriseOne Product Packaging Tools. If any fields use a token, you can convert the token into an absolute path or you can convert the absolute path into a token. For example, you can convert any instance of \$DEP into Z:\DepServer or you can convert Z:\DepServer into \$DEP.

An absolute path name begins with the root directory and details the entire path, such as Z:\DepServer\CD Templates\Partial Update. A relative path name begins with a token, such as \$DEP. The token represents at least the root directory and possible contiguous directory paths. For example, \$DEP might represent Z:\DepServer. This relative path might appear within an applicable field as \$DEP\CD Templates\Partial Update.

You do not need to define the tokens. The JD Edwards EnterpriseOne Product Packaging Tool automatically resolves the token values based on the installation and other information that is available to it.

These are the relative path tokens that you can use:

#### **\$DEP**

Represents the root directory where JD Edwards EnterpriseOne is installed on the deployment server.

#### **\$IMAGE**

Represents the image path of the software master.

#### **\$CLIENT**

Represents the root directory where JD Edwards EnterpriseOne is installed on the workstation that you are currently using.

#### **\$TEMP**

Represents the Microsoft Windows temporary directory of the workstation that you are currently using.

## 7.2.2 Forms Used to Work With Relative and Absolute Paths

Form Name	FormID	Navigation	Usage
Work With Software Mastering	W9640B	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Product Packages (P9640) . In the Work With Software Mastering form, select Convert Paths from the Row menu.	Used to delete a data cleanup template.
Report Output Destination	W98UBEA	In the Report Output Destination form, click OK.	Used to select the report destination and output stream access.

## 7.2.3 Using Relative Path Tokens

To use relative path tokens:

From a field that allows relative paths, enter a token followed by the remainder of the path that is needed for that field.

For example:

```
$DEP\CD Templates\Partial Update
```

## 7.2.4 Converting Relative and Absolute Paths

The software converts the absolute and relative paths for any fields within the JD Edwards EnterpriseOne Product Packaging Tools to which this applies.

Access the Work With Software Mastering form.

**Figure 7-6 Product Packages [Work with Software Mastering] form**

Release	Build Phase	Build Phase Description	CD Type	Type Description	CD Name	Description

1. Select the master for which you want to convert relative and absolute paths, and then select Convert Paths from the Row menu.
2. In Report Output Destination, select the option to send the report to a printer or to the screen, and then click OK.

The software converts the absolute and relative paths for any fields within the JD Edwards EnterpriseOne Product Packaging Tools to which this applies.

## 7.3 Working with Data Cleanup Procedure Templates

This section discusses how to:

- Work with Data Cleanup Procedure Templates
- Add a Data Cleanup procedure
- Copy a Data Cleanup procedure

### 7.3.1 Forms Used with Data Cleanup Procedure Templates

Form Name	FormID	Navigation	Usage
Work with Procedure	W9646A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Advanced Operations, Data Cleanup Procedure Templates (P9646)	Used to execute or delete a data cleanup procedure.
Procedure Definition Director	W9646B	In the Work with Procedure form, click Add.	Takes you through the steps to add a data cleanup procedure.
Add Procedure	W9646C	In the Add Procedure form, enter the information and click Next.	Used to enter information about a data cleanup procedure.
Procedure Revision	W9646D	In the Work with Procedure form, select a procedure.	Used to revise or execute a data cleanup procedure.
Copy Procedure	W9646L	In the Work with Procedure form, select a procedure and click Copy.	Used to create a new data cleanup procedure based on an existing one.

### 7.3.2 Setting Up Data Cleanup Procedure Templates

Although this JD Edwards EnterpriseOne Product Packaging Tool function is primarily used to create data cleanup procedures, you can use it to create any type of procedure to run a batch process or Microsoft Windows executable.

Access the Work With Procedure form.

#### Release

The system release number to which the data cleanup applies.

#### Name

The unique name given to a data-cleanup procedure.

**Description**

A short description of the data-cleanup procedure.

**Status**

The status of the data cleanup procedure. Indicates where the data cleanup procedure resides in the process between defining and executing. For example, the procedure could be at a status of defined, validated, or built. Before executing the data-cleanup procedure (whether from the software master or from the data cleanup procedure application), verify that the status of the procedure, and all of its build steps that you want to run with it, are at validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Type**

The type of procedure being run.

**Comment**

A short informational comments provided by the system about each step of the mastering process. You can enter our own comments in this field, but the system will over write any text in the field during the mastering process.

**User ID**

The code that identifies the user profile.

### 7.3.3 Adding a Data Cleanup Procedure

This function enables you to add procedure information and the steps that are necessary to clean up data.

#### 7.3.3.1 Procedure Definition Director

The JD Edwards EnterpriseOne Procedure Definition Director guides you through the steps necessary to define a procedure.

Access the Procedure Definition Director form.

#### 7.3.3.2 Add Procedure

The Add Procedure is used to enter information about the new procedure. You will need to enter the unique name and other information related to the procedure.

Access the Add Procedure form.

**Name**

The unique name given to a data cleanup procedure.



**Description**

A short description of the data cleanup procedure.

**Comment**

A short information comment that you provide about the data cleanup procedure.

**Status**

The status of the data cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. For example, the procedure could be at a status of defined, validated, or built. Before executing the data cleanup procedure (whether from the software master or from the data cleanup procedure application), verify that the status of the procedure, and all of its steps that you want to run with it, are at validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Procedure Release**

The system release number to which the data cleanup applies.

**7.3.3.3 Procedure Revision**

The Procedure Revision is used to revise or execute a data cleanup procedure.

Access the Procedure Revision form.

**Name**

The unique name given to a data cleanup procedure.

**Description**

A short description of the data cleanup procedure.

**Status**

The status of the data cleanup procedure. Indicates where the data-cleanup procedure resides in the process between defining and executing. For example, the procedure could be at a status of defined, validated, or built. Before executing the data-cleanup procedure (whether from the software master or from the data cleanup procedure application), verify that the status of the procedure, and all of its steps that you want to run with it, are at validated (30). Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Comment**

A short information comment that you provide about the data cleanup procedure.

**Release**

The system release number to which the data cleanup applies.

### 7.3.4 Copying a Data Cleanup Procedure

The Copy Procedure is used to copy one procedure to another.

Access the Copy Procedure form.

**Copy From Procedure**

The unique name given to the data cleanup procedure.

**Copy To Procedure**

The unique name given to the data cleanup procedure.

## 7.4 Viewing a Table Change Table

The change table is table that contains all the source table data changes and system metadata necessary to maintain the change table. This function allows you to view the Table Change table.

### 7.4.1 Forms Used to Work with Table Change Table

Form Name	FormID	Navigation	Usage
Work With Table Change Table	W9698710A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Advanced Operations, View Table Change table (P9698712)	Used to view the Table Change table.

### 7.4.2 Viewing the Table Change Table

Access the Work With Table Change Table form.

Figure 7-7 View Table Change Table [Work with Table Change Table] form

Table Name	From Release	To Release	Table Change Type	Table Change Type	EnterpriseOne Only (Y/N)	User ID	Program ID	Date Updated	Work Stn ID
------------	--------------	------------	-------------------	-------------------	--------------------------	---------	------------	--------------	-------------

**Table Name**

The name that identifies a system object. JD Edwards EnterpriseOne architecture is object-based. Discrete software objects are the building blocks for all applications and developers can reuse the objects in multiple applications. The Object Librarian tracks each object.

**From Release**

The release number as defined in the Software versions Repository file.

**To Release**

The release number as defined in the Release Master.

**Table Change Type**

Describes the type of action taken on a table or index during the upgrade or update process.

**EnterpriseOne Only (Y/N)**

This field is used to describe the table associated with a scheduled conversion program. It is used to determine if the table is used by the system software only or if it is shared by JD Edwards EnterpriseOne software.

**User ID**

The code that identifies a user profile.

**Program ID**

The number that identifies the batch or interactive program (batch or interactive object).

The program ID is a variable length value. It is assigned according to a structured syntax in the form TSSXXX where:

**T** - The first character of the number is alphabetic and identifies the type.

**SS** - The second and third characters of the number are numeric and identify the system code.

**XXX** - The remaining characters of the number are numeric and identify a unique program or report.

**Date Updated**

The date that specifies the last update to the file record.

**Work Stn ID**

The code that identifies the work station ID that executed a particular job.

**Time Updated**

The time that specifies when the program executed the last update to this record.

## 7.5 Viewing the Index Change Tables

This function allows you to view the index change tables.

## 7.5.1 Forms Used to Work with Index Change Table

Form Name	FormID	Navigation	Usage
Work With Index Change Table	W9698712A	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Advanced Operations, View Index Change Table (P9698712)	Used to view the Index Change Tables.

## 7.5.2 Viewing the Index Change Tables

Access the Work With Index Change Table form.

**Figure 7–8 View Index Change Table [Work with Index Change Table] form**

Table Name	Index Identifier	From Release	To Release	Index Change Type	Index Change Type	EnterpriseOne Only (Y/N)	User ID	Program ID	Up
------------	------------------	--------------	------------	-------------------	-------------------	--------------------------	---------	------------	----

**Table Name**

The name that identifies a system object. JD Edwards EnterpriseOne architecture is object-based. Discrete software objects are the building blocks for all applications and developers can reuse the objects in multiple applications. The Object Librarian tracks each object.

**Index Identifier**

The numeric ID used for the table's index.

**From Release**

The release number as defined in the Software Versions Repository file.

**To Release**

The release number as defined in the Release Master.

**Index Change Type**

Describes the type of action taken on a table or index during the upgrade or update process.

**EnterpriseOne Only (Y/N)**

This field is used to describe the table associated with a scheduled conversion program. It is used to determine if the table is used by the system software only or if it is shared by JD Edwards EnterpriseOne software.

**User ID**

The code that identifies a user profile.

**Program ID**

The number that identifies the batch or interactive program (batch or interactive object).

The program ID is a variable length value. It is assigned according to a structured syntax in the form TSSXXX where:

**T** - The first character of the number is alphabetic and identifies the type.

**SS** - The second and third characters of the number are numeric and identify the system code.

**XXX** - The remaining characters of the number are numeric and identify a unique program or report.

**Date Updated**

The date that specifies the last update to the file record.

**Work Stn ID**

The code that identifies the work station ID that executed a particular job.

**Time Updated**

The time that specifies when the program executed the last update to this record.

## 7.6 Viewing the Control Change Tables

Control Change Tables are generated during the installation process when you run the environment workbench. This function allows you to view the Control Change tables.

## 7.6.1 Forms Used with Software Update Control Tables

Form Name	FormID	Navigation	Usage
Software Update Control Tables	W96470I	In Solution Explorer, navigate to System Administration Tools, Package and Deployment Tools, Product Packaging, Advanced Operations, View Control Change Tables (P96470)	Used to add change or delete a software update control tables.

## 7.6.2 Viewing the Software Update Control Tables

A software update control table is a table that contains the items that have changed between releases or updates.

Access the Software Update Control Tables form.



**Figure 7–9 View Control Change Tables [Software Update Control Tables] form**

View Control Change Tables - [Software Update Control Tables]

File Edit Preferences Form Window Help

Select Find Close Seq... N Links Copy... OLE... Internet

From Release

To Release

Control Tables in an Update

=Add

=Change

=Delete

**From Release**

The current release level of your JD Edwards software before you upgrade it.

**To Release**

The release number as defined in the release master.



---

## Working with Translation Updates

This chapter contains the following topics:

- [Section 8.1, "Understanding Translation Updates"](#)
- [Section 8.2, "Creating Translation Update Templates"](#)
- [Section 8.3, "Collecting Translation Updates"](#)
- [Section 8.4, "Creating an Update Package"](#)

### 8.1 Understanding Translation Updates

When an Oracle's JD Edwards EnterpriseOne software update is taken by a customer that uses a language not supported by JD Edwards EnterpriseOne, that customer's translated database requires updating. In addition, when a new translation is made for a customer that uses an unsupported language, the new translation must be deployed to the customer's database. JD Edwards EnterpriseOne Product Packaging Tool provides a way for business partners who support customers that use languages not offered by JD Edwards EnterpriseOne to create language updates, wrap the language updates into a database, and then create an update package to deliver the translation updates to customers. You use the Solution Explorer to perform the tasks for translation updates.

The JD Edwards EnterpriseOne Product Packaging Tool is a scripting engine that provides templates. Each template is a script that is used to build an update package. The update packages are deployed to the customers' systems. After you make a language update and check it into a JD Edwards EnterpriseOne OMW project as an object, you run the Language Update Template UBE (R95160) to create a copy of the template on your JD Edwards EnterpriseOne system. After you create the template, you use the Language OMW Groupings Program (P9649) to collect the translation update objects from OMW and wrap them into a database. You use the Work With CD Configurations Program (P9640) to create a package build, which is an executable file that can be deployed to customers' databases.

---

**Note:** The Translation Update Template is not intended to be used to build an entire language CD for updates.

---

### 8.2 Creating Translation Update Templates

This section provides an overview of the Translation Update Template and discusses how to create the Translation Update Template.

## 8.2.1 Understanding Translation Update Templates

Language templates are available with the JD Edwards EnterpriseOne Product Packaging Tool. You create the Translation Update Template by running Oracle's JD Edwards EnterpriseOne Insert Language Update Template (R95160) UBE. This UBE has processing options that enable you to specify the release and data source for the template. When the UBE successfully runs, you receive a report indicating header, details, build steps, and control record were successfully inserted.

## 8.2.2 Prerequisites

The JD Edwards EnterpriseOne OMW project contains objects for which you have already made translations.

## 8.2.3 Setting Processing Options for the Translation Update Template (R95160)

Processing options enable you to specify the release and data source for the template.

### 8.2.3.1 Template Options

Use these processing options to specify the release for which the template is to be used and to specify where the template should be inserted. Use the processing option for **Release** to specify the JD Edwards EnterpriseOne release for which the template is to be used. You must provide a value for this processing option. If you leave the processing option for **Release** blank, the template will not be created. Use the processing option for **Data Source** to specify the destination for the Translation Update Template. If you leave this processing option blank, the system uses the OCM setting to determine where to insert the template.

#### Release

Specify the release for the Language Update Template. The release value must be the same as the release of the project that you are using, and this value is required for the batch application to run. Use the visual assist to locate the release.

#### Data Source

Specify the data source where the Language Update Template should be inserted. If you leave this field blank, the system determines the destination of the template from OCM. Use the visual assist to specify a data source.

## 8.3 Collecting Translation Updates

This section lists the forms to collect translation updates and discusses how to build a database of the translation updates.

### 8.3.1 Forms Used to Collect Translation Updates

Form Name	FormID	Navigation	Usage
Work with Language Groups	W9649A	In the Fast Path field of Solution Explorer, type P9649 .	Locate and copy the language that is to be mastered for the new project.
Language Group Information	W9649B	Select Copy on the Work with Language Groups form.	Enter information to create the database.

Form Name	FormID	Navigation	Usage
Work with OMW Project Detail	W9649F	Click Next on the Language Group Information form.	Select the OMW Project where the language updates have been checked in as objects.
OMW Project Search & Select	W9649G	Click Browse on the Work with OMW Project Detail form.	Search for and Select the OMW Project that includes the objects for the language updates.
Work With OMW Project Detail	W9649F	Click Close on the OMW Project Search & Select form.	Select the OMW Project that includes the objects for the language updates.
Select Languages	W9649C	Click Next on the Work with OMW Project Detail form.	Select the language and data source for which you want to build a database.
Language Information	W9649D	Click Copy on the Select Languages form.	Enter language and data source mapping data source information.
Work with Language Groups	W9649A	Click Finish on the Select Languages form or click OK on the Language Information form.	Build a database that contains the translation updates.

## 8.3.2 Creating a Translation Updates Database

After you enter Oracle's JD Edwards EnterpriseOne Language OMW\_Groupings (P9649) program, each of these forms automatically appears as a result of selecting an option on the current form. The forms appear in the order provided in the Process Introduction Table.

### 8.3.2.1 Work With Language Groups

The Work with Language Groups is used to select the language that is to be mastered for the project. This form appears the first time that the record is created; it does not appear if you are inspecting the record on an update. If you need to change the language or mapping information, you can access this form by clicking Add or Copy on the Select Languages form

Access the Work with Language Groups form.

#### Language Group

The language group name.

#### Description

A user defined name or remark.

#### Database File Path

An enterprise directory path that you specify, which will be the root directory containing the contents of your software master.

**OMW Project Type**

An identifier for an enterprise project. An enterprise project is composed of a group of enterprise objects that have been modified or created by a developer to complete a task. All work with objects within enterprise must occur within the context of a project.

**Language**

The language name identifier.

**8.3.2.2 Language Group Information**

The Language Group Information form is used to enter the name and description for the language group and the directory path where the language database(s) will be stored. Use the \*ALL option to select if whole language is to be mastered or Project if the OMW project names are associated with this language group.

Access the Language Group Information form.

**Language Group**

The language group name.

**Description**

A user defined name or remark.

**Data File Path**

An Enterprise directory path that you specify which will be the root directory containing the contents of your software master.

**All**

Select All if whole language is to be mastered.

**Project**

Select Project if the OMW projects names are to be associated with this language.

**8.3.2.3 Work with OMW Project Detail**

The Work with OMW Project Detail form is used to select the individual projects to be used in the language group.

Access the Work with OMW Project Detail form.

**OMW Project Name**

An identifier for an Enterprise project. An Enterprise project is composed of a group of Enterprise objects that have been modified or created by a developer to complete a task. All work with objects within Enterprise must occur within the context of a project.

**Description**

A user defined name or remark.

**8.3.2.4 OMW Project Search & Select**

The OMW Project Search & Select form is used to display a list of projects to select from when you click on the browse button on the Work with OMW Project Detail form.

Figure 8–1 P9649 [OMW Project Search &amp; Select] form

	OMW Project Name	Description	Project Status	Project Type	Project Severity	Product Code Reporting	Date Entered
4095262		Tom Chen Test Project	21	02	04	H95	02/23/2005
5253859		Create Batch to print tasks	26	01	02	17C	02/23/2005
5692190		R48425 Print Customer number	01	01	03	13	01/25/2005
5706643		Sort feature on P1206	01	01	02	13	01/31/2005
6030438		P48011 SUB search by MCU	01	01	02	13	01/31/2005
6300830		Combine Maint Loops w/Assoc PM	01	01	03	13	02/02/2005
6491015		WO page numbering for R48425	01	01	03	13	01/25/2005
6501367		Service Type Description	01	01	02	13	01/31/2005
6562813		Search by Requested Date DRQJ	01	01	03	13	01/19/2005
6579025		Media Object Issues	01	01	02	17	02/04/2005
6579033		Media Object Issues	01	01	02	17	02/07/2005
6911868		Increase Media Object fields	26	01	02	17	02/07/2005
7012901		Add Events to QE Environments	91	02	03	H99Q	01/31/2005
7048496		Cannot add supplier on the fly	01	01	03	43	01/10/2005
7052153		P4074 - Exchange Rate in PO	01	01	03	43	01/05/2005
7053800		P04571 -field VLDT not update	21	01	03	04	01/21/2005
7055451		Error in LT Field on Inquiry	21	01	03	09	01/20/2005
7095321		Alternate Address Edit	01	01	02	01	01/09/2005
7155101		Can't Select 1st Grid Row	01	01	02	15	01/06/2005
7159524		ERROR ON ADD/COPY FORM	01	01	03	F30L	01/20/2005
7163654		Error increments line number	01	01	03	43	01/06/2005
7166441		Copy Function on Detail	01	01	02	15L	01/05/2005
7166759		Dependency Checker/Token Inher	21TD	02	02	H95	01/19/2005
7184560		Item Ledger & Batch Doc-Type	21	01	02	42	03/07/2005
7185600		Partial Shipment Validation	21	01	02	49	03/01/2005
7190222		PC MILER Memory Leak	20	01	02	49	02/08/2005

**OMW Project Name**

An identifier for a JD Edwards EnterpriseOne project. A JD Edwards EnterpriseOne project is composed of a group of JD Edwards EnterpriseOne objects that have been modified or created by a developer to complete a task. All work with objects within JD Edwards EnterpriseOne must occur within the context of a project.

**Description**

A user defined name or remark.

**Project Status**

A value that indicates the status of a project. Values are:

01 Complete

11 New Project Pending Review

21 Programming

25 Rework - Same Issue

26 QA Test/Review

28 QA Test/Review Complete

38 In Production

40 Production Development

41 Production to Prototype

42 Transfer Prototype to Development

45 Pristine Get

91 Cancelled-Entered in Error

This project status should be the starting point in the status activity rules that are defined in the Object Management Configuration Application (P98230).

**Project Type**

An identifier for a JD Edwards EnterpriseOne project. A JD Edwards EnterpriseOne project is composed of a group of JD Edwards EnterpriseOne objects that have been modified or created by a developer to complete a task. All work with objects within JD Edwards EnterpriseOne must occur within the context of a project.

**Project Severity**

The criticality of the project.

**Product Code Reporting**

A user defined code (98/SY) that specifies the system number for reporting and jargon purposes.

**Date Entered**

The date the project was entered.

**Date Started**

The date the project was started.

**Planned Completion**

The date the project is planned for completion

**Completion Date**

The actual date the project was completed.

**Order Number**

A number that identifies the original document. This document can be a voucher, a sales order, an invoice, unapplied cash, a journal entry, and so on.



**Category Code 1**

Future use.

**Category Code 2**

Future use.

**Category Code 3**

Future use.

**Category Code 4**

Future use.

**Category Code 5**

Future use.

**Category Code 6**

Future use.

**Category Code 7**

Future use.

**Category Code 8**

Future use.

**Category Code 9**

Future use.

**Category Code 10**

Future use.

**Program ID**

The number that identifies the batch or interactive program (batch or interactive object).

The program ID is a variable length value. It is assigned according to a structured syntax in the form TSSXXX where:

**T** - The first character of the number is alphabetic and identifies the type.

**SS** - The second and third characters of the number are numeric and identify the system code.

**XXX** - The remaining characters of the number are numeric and identify a unique program or report.

**Machine Key**

The Location or Machine Key indicates the name of the machine on the network (server or workstation).

**User ID**

The code that identifies the user profile.

**Date Updated**

The date that specifies the last date to the file record.

**Time Updated**

The time that specifies when the program executed the last update to this record.

**Time Zone**

Choose the time zone for which you want to view the date and time.

**Rule Name**

Unique name identifying a daylight savings rule. Use daylight savings rules to adjust time for a geographic and political locale.

**Source Release Number**

The transfer rule only applies to objects in the project for this release (objects are always tied to a release within a project). The From and To Release fields should always be equal.

**Target Release Number**

The transfer rule only applies to the objects in the project that are for this release. (Objects are always tied to release within a project). The From and To release fields should always be equal.

### 8.3.2.5 Select Languages

The Select Languages form is used to select the language and data source where the language records for the specified language resides.

Access the Select Language form.

**L**

A user defined code (01/LP) that specifies the language to use on forms and printed reports. before you specify a language, a code for that language must exist at either the system level or in your user preference.

**Language**

A user defined code (01/LP) that specifies the language to use on forms and printed reports. before you specify a language, a code for that language must exist at either the system level or in your user preference.

**Data Source**

The name that identifies the data source.

**Media Objects Data Source**

A user defined name or remark.

## 8.4 Creating an Update Package

This section lists the forms to create an update package and discusses how to create an update package.

### 8.4.1 Understanding Package Updates

After you enter the Work With CD Configuration (P9640) program, each of these forms automatically appears as a result of selecting an option on the current form. The forms appear in the order listed in the Process Instruction Table. The Build Steps tab on the CD Revisions form enables you to select the new database that you created in the previous task, and then create an update package that is sent to your workstation.

### 8.4.2 Prerequisites

Before you complete the task in this section:

- Verify that the Language Update Template is available.
- Verify that the database for the translation updates is available.

### 8.4.3 Forms Used to Create an Update Package

Form Name	FormID	Navigation	Usage
Work With Software Mastering	W9640B	In Solution Explorer, navigation to System Administration Tools, Package and Deployment Tools, Product Packaging, Product Packages (P9640)	Add a new update package.
CD Configuration Director	W9640D	Click Add on the Work With Software Mastering form.	Create a CD beginning page.
CD Information	W9640H	Click Next on the CD Configuration Director form.	Create the update package build.
CD Revisions	W9640G	Click Next on the CD Information form.	Enter location information for a valid CD and the OMW project.
Work with Language Groups	W9649A	Click the Value button from the Build Steps tab on the CD Revisions form.	Select the language package that you created in the Collecting Translation Updates task.

### 8.4.4 Creating an Update Package

After the build is finished, verify that the executable file was created and is on your workstation in the location that you specified.

#### 8.4.4.1 Work With Software Mastering

Access the Work With Software Mastering form.

##### Release

The system release number of the software master.

##### Build Phase

This is the phase of development that your software master represents, such as an alpha, a beta, or a general availability phase. You can add or changes the phases since this is a user defined code. Values are:

01 Alpha

02 Beta

03 General Availability

##### Build Phase Description

A user defined name or remark.

##### CD Type

The type of CD being mastered.

**Type Description**

A user defined name or remark.

**CD Name**

A unique name given to a software master.

**Description**

A short description of the software master.

**Status**

The status of the software master within the mastering process. Values are:

10 In Definition

20 Defined

30 Validated

40 Processing

50 Failed

60 Built

70 Obsolete

80 Mastered

90 Repaired

**Status Description**

A user defined name or remark.

**Comment**

A user defined name or remark.

**Definitions Date**

The date the CD was defined.

**CD Master Date**

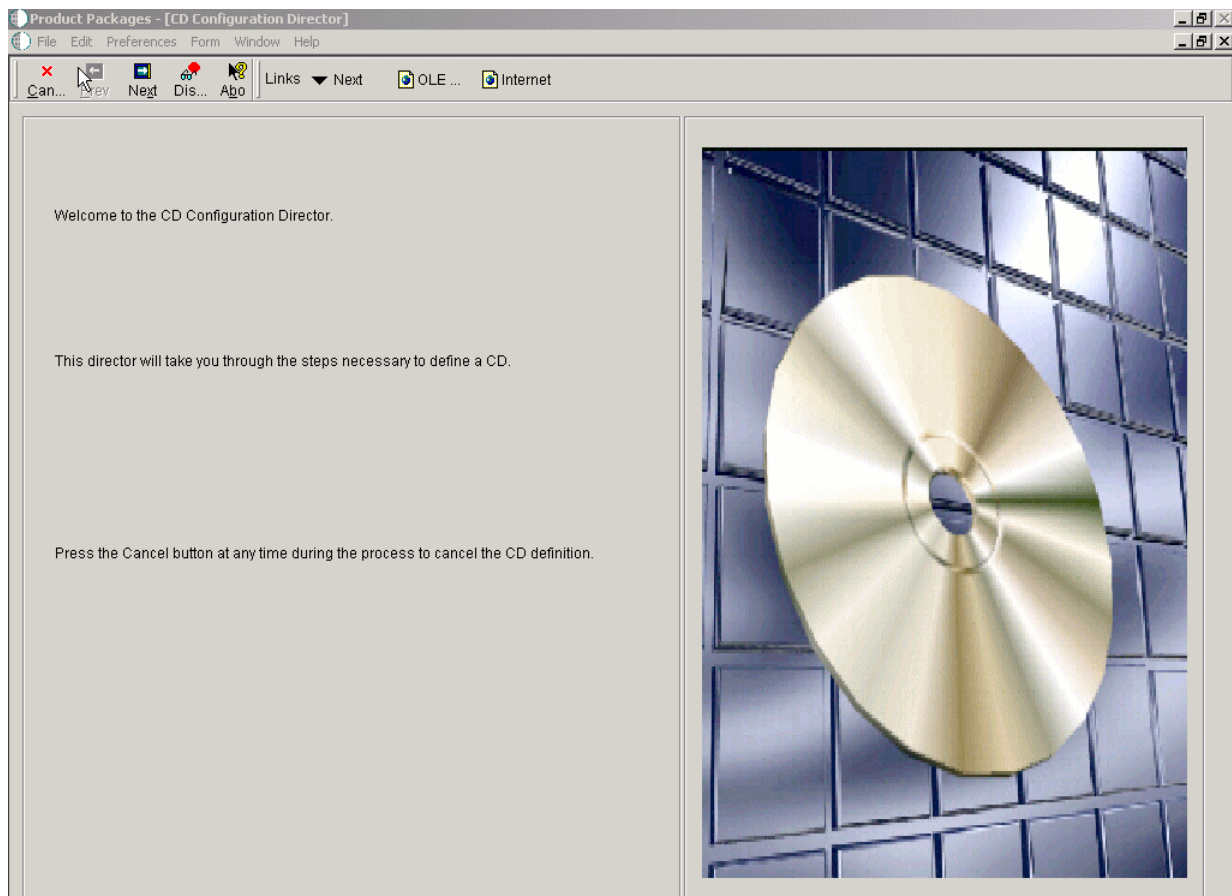
The date the CD was mastered.

**One Off CD**

Future use.

**8.4.4.2 CD Configuration Director**

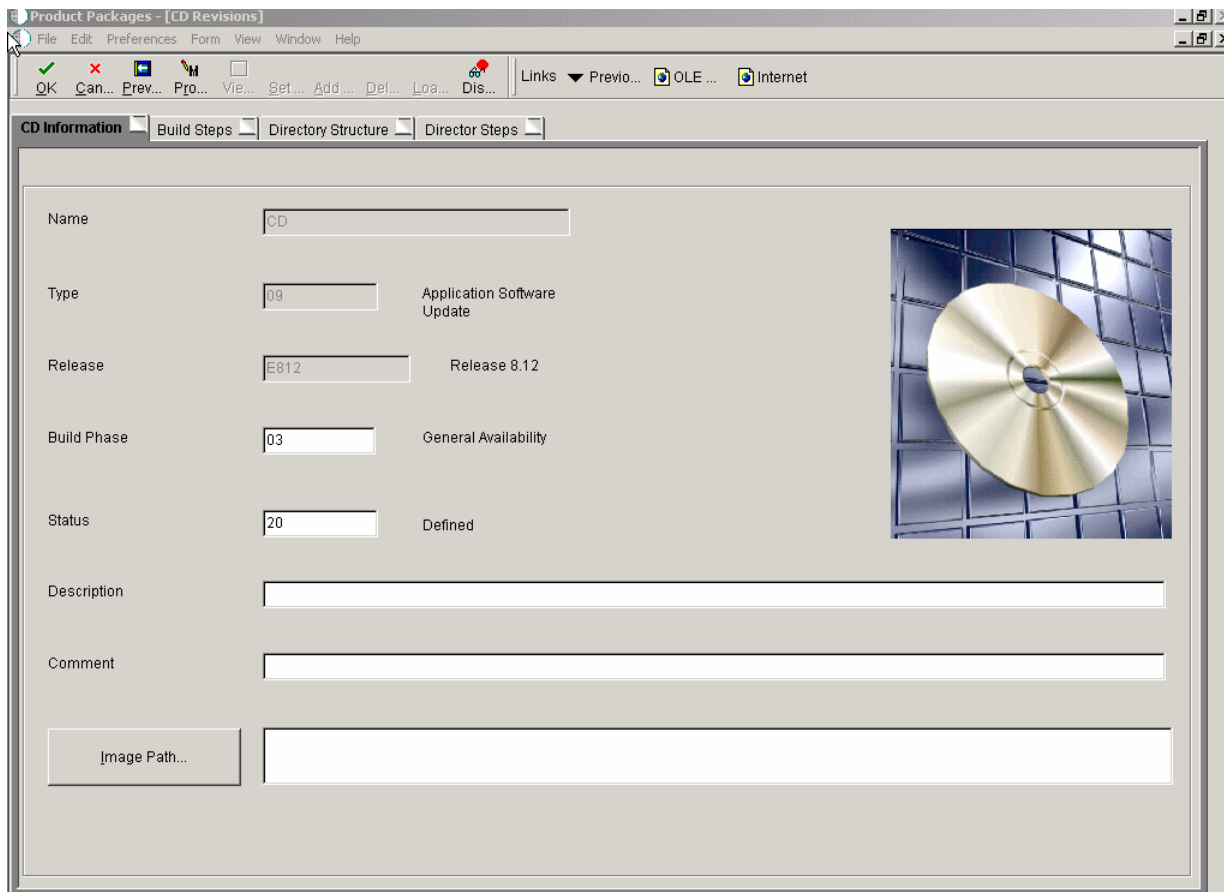
The JD Edwards EnterpriseOne CD Configuration Director guides you through the steps necessary to define a CD.

**Figure 8-2** *Product Packages [CD Configuration Director] form*

#### 8.4.4.3 CD Revisions

The CD Information tab lets you set up your CD information, build steps, directory structure, and director steps.

**Figure 8–3 Product Packages [CD Information] form**



The screenshot shows a software application window titled "Product Packages - [CD Revisions]". The window has a menu bar with "File", "Edit", "Preferences", "Form", "View", "Window", and "Help". Below the menu bar is a toolbar with buttons for "OK", "Can...", "Prev...", "Pro...", "View...", "Set...", "Add...", "Del...", "Loc...", "Dis...", "Links", "Previo...", "OLE ...", and "Internet". The main area of the window has four tabs: "CD Information", "Build Steps", "Directory Structure", and "Director Steps". The "CD Information" tab is selected. It contains the following fields and labels:

- Name:** A text box containing "CD".
- Type:** A text box containing "09" and a label "Application Software Update".
- Release:** A text box containing "E812" and a label "Release 8.12".
- Build Phase:** A text box containing "03" and a label "General Availability".
- Status:** A text box containing "20" and a label "Defined".
- Description:** An empty text box.
- Comment:** An empty text box.
- Image Path...:** A button next to a large empty text box.

On the right side of the form, there is a small image of a CD on a blue grid background.

1. Enter your CD Information.

- **Status**

Displays the status of a build step or directory structure within the mastering process. For example, the build step or directory structure could be at a status

of defined, validated, or built. Use the visual-assist button to view a complete list of values for this field.

- **Comment**

Displays short informational comments that are provided by the system about each step of the mastering process. You can enter comments into this field, but the system will overwrite any text that is currently in the field during the next mastering process.

- **Image Path**

Overfills the directory path where you want the software master to reside on the enterprise. The directory that you indicate will be the root directory for the creation of the final software master.

## 2. Define your Build Steps.

- **Sequence**

Displays the order in which the system processes build steps, usually in increments of 10 (for example, **10**, **20**, **30**, and so on).

Verify the order in which you want the step performed during the creation of the software master definition. You can use the same sequence number for multiple build steps.

- **Type**

Displays the type of CD being mastered. Verify which component type you want the build step to perform. For example, you can define steps to create file structures, build a package, or create INF files.

- **Status**

Verifies the current status of the build step, which can indicate whether to run the particular step during the mastering process. For example, if you need to perform a manual step during the mastering process, you can manually complete the step, change the step's status to **60** (Built) by choosing the step, and then, from the View menu, select Execute Step.

- **Target Folder**

Displays the name of the target folder, such as **planner** or **updatepkg**. The system populates this field based upon the template that was used to define the software master. Change the name of the target folder only if you added the folder yourself.

- **Executable**

Displays the software that might populate this field based upon the build step's definition in the software master template. To change this field, click the **Executable** button and use the Select a Windows Executable form to select an executable for this build step.

- **Value/Parameter**

Verifies the value or parameter. To change this information, click the **Value/Parameter** button, and then enter a value or parameter that is appropriate to the step you chose. This field is dynamic. The button name is either Value or Parameter, based upon the step that you select. The form that appears when you click the button also depends upon which step you select. For example, the Package Build step calls the Select Package Build form, from which you can select a package; and the Build Change Tables step calls the

Work With Change Table Definitions form, from which you can modify the change table configuration.

3. Set up your Directory Structure.

■ **Source Path**

Displays the directory on the enterprise that you want to use to populate the highlighted target folder. You can use an absolute (full) directory path name, or you can use a relative path name with a token. The system copies the data that is contained in the source path, depending upon how you filter those files (see the **File Filter** field description for this information), into the highlighted target folder of the software master.

Verifies the directory path of the source that will populate the component. This is the source path name of the data that you want to retrieve for the master and that you placed into the target folder. You can specify this path using either a relative or absolute path. See [Appendix A, "Troubleshooting Product Packaging"](#) for information about relative and absolute paths.

■ **File Filter**

Displays the filter, which you can designate, for the files that are contained in the source path. For example, you can use \*.\* to retrieve all files in the source path, or \*.exe to retrieve only executable programs.

Verify the file filter. The default is \*.\*. The file filter enables you to filter any data files that you retrieve from the source directory. For example, you can use \*.\* to retrieve all files in the source directory.

■ **Full Path**

Displays the names and functions. This field is dynamic and its name and function depend upon the build step that is highlighted. The possible names and functions are:

**Value:** The value that is needed to complete the highlighted build step. For example, if the build step is a batch process, the value field contains the name of the batch process and its version, such as **R95012\XJDE0001**.

**Parameter:** The parameters that are needed for the highlighted build steps that process an executable program.

The full (absolute) target path name of the highlighted directory, such as **Root/planner/updatepkg**.

4. Define your Director Steps.

■ **Package/Change Table Form**

Indicates the package form and change table form used.

■ **Software Update Form**

Future use.



---

# Troubleshooting Product Packaging

This appendix discusses how to troubleshoot:

- [Section A.1, "Failure During the Check Mastering Items Step"](#)
- [Section A.2, "Failure to Build the Feature INF File"](#)
- [Section A.3, "Failure to Copy Change Tables"](#)
- [Section A.4, "Failure to Copy Custom Tables"](#)
- [Section A.5, "Unable to Find Product Package After Installation"](#)
- [Section A.6, "Problems with Object Librarian Records"](#)

## A.1 Failure During the Check Mastering Items Step

When mastering the CD, the process fails during the Check Mastering Items step.

### A.1.1 Solution

This step is a manual build; it must be performed manually. Highlight the step and select Execute Step from the View exit menu. After the step is executed, click Master CD and then proceed.

## A.2 Failure to Build the Feature INF File

JD Edwards EnterpriseOne Product Packaging mastering process fails at the build step Copy Package and Feature INFs - report R9600400E. Because features are a part of the Package Build process, update packages do not build feature INF files.

### A.2.1 Solution

If features are included in the chosen update package, be sure that the Feature INF files were created previously.

## A.3 Failure to Copy Change Tables

No change tables (for instance, F960004 and F960005) exist in the XML database, and the jde.log file lists several errors indicating that it cannot find these tables in the Control tables. This problem indicates that the change tables do not exist in the mastering environment.

### **A.3.1 Solution**

Verify that the tables exist and are set up correctly in the mastering environment.

## **A.4 Failure to Copy Custom Tables**

Custom tables were not copied to the XML database, even though these tables were included in the update package and exist in the environment that the package is built from. This problem is an indication that the source environment that is specified in Oracle's JD Edwards EnterpriseOne Change Table Configurator is not set up correctly or that the table exists in that environment.

### **A.4.1 Solution**

Verify that the specified environment is valid and has tables associated with the environment, and that the custom table cannot be found in that environment.

## **A.5 Unable to Find Product Package After Installation**

After you run the install manager to install the Software Master on the new system, the product package is not listed in Oracle's JD Edwards EnterpriseOne Software Updates application. This error occurs because the package name does not match the product package name.

### **A.5.1 Solution**

Remaster the Software Master and ensure that the update package name is the same as the product package name. Also, make sure that you have Oracle's latest JD Edwards EnterpriseOne Planner update installed.

## **A.6 Problems with Object Librarian Records**

A potential problem exists for objects that belong to a project in the original system that was created with a SAR that contains an alpha character. This problem would be evident if you try to enter that project after installation on the destination system.

### **A.6.1 Solution**

After the package is installed on the destination deployment server, open the created JD Edwards EnterpriseOne Product Packaging XML database and modify the F9861 table. In the SIMSAR column, make sure that no records have alpha characters for values. Change all alpha characters to 0.

---

---

# Glossary

## **Accessor Methods/Assessors**

Java methods to “get” and “set” the elements of a value object or other source file.

## **activity rule**

The criteria by which an object progresses from one given point to the next in a flow.

## **add mode**

A condition of a form that enables users to input data.

## **Advanced Planning Agent (APAg)**

A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of relational databases, flat file format, and other data or message encoding, such as XML.

## **application server**

Software that provides the business logic for an application program in a distributed environment. The servers can be Oracle Application Server (OAS) or WebSphere Application Server (WAS).

## **Auto Commit Transaction**

A database connection through which all database operations are immediately written to the database.

## **batch processing**

A process of transferring records from a third-party system to JD Edwards EnterpriseOne.

In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.

## **batch server**

A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.

**batch-of-one**

A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks.

**best practices**

Non-mandatory guidelines that help the developer make better design decisions.

**BPEL**

Abbreviation for Business Process Execution Language, a standard web services orchestration language, which enables you to assemble discrete services into an end-to-end process flow.

**BPEL PM**

Abbreviation for Business Process Execution Language Process Manager, a comprehensive infrastructure for creating, deploying, and managing BPEL business processes.

**Build Configuration File**

Configurable settings in a text file that are used by a build program to generate ANT scripts. ANT is a software tool used for automating build processes. These scripts build published business services.

**build engineer**

An actor that is responsible for building, mastering, and packaging artifacts. Some build engineers are responsible for building application artifacts, and some are responsible for building foundation artifacts.

**Build Program**

A WIN32 executable that reads build configuration files and generates an ANT script for building published business services.

**business analyst**

An actor that determines if and why an EnterpriseOne business service needs to be developed.

**business function**

A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules, and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.

**business function event rule**

See named event rule (NER).

**business service**

EnterpriseOne business logic written in Java. A business service is a collection of one or more artifacts. Unless specified otherwise, a business service implies both a published business service and business service.

**business service artifacts**

Source files, descriptors, and so on that are managed for business service development and are needed for the business service build process.

**business service class method**

A method that accesses resources provided by the business service framework.

**business service configuration files**

Configuration files include, but are not limited to, interop.ini, JDBj.ini, and jdelog.properties.

**business service cross reference**

A key and value data pair used during orchestration. Collectively refers to both the code and the key cross reference in the WSG/XPI based system.

**business service cross-reference utilities**

Utility services installed in a BPEL/ESB environment that are used to access JD Edwards EnterpriseOne orchestration cross-reference data.

**business service development environment**

A framework needed by an integration developer to develop and manage business services.

**business services development tool**

Otherwise known as JDeveloper.

**business service EnterpriseOne object**

A collection of artifacts managed by EnterpriseOne LCM tools. Named and represented within EnterpriseOne LCM similarly to other EnterpriseOne objects like tables, views, forms, and so on.

**business service framework**

Parts of the business service foundation that are specifically for supporting business service development.

**business service payload**

An object that is passed between an enterprise server and a business services server. The business service payload contains the input to the business service when passed to the business services server. The business service payload contains the results from the business service when passed to the Enterprise Server. In the case of notifications, the return business service payload contains the acknowledgement.

**business service property**

Key value data pairs used to control the behavior or functionality of business services.

**Business Service Property Admin Tool**

An EnterpriseOne application for developers and administrators to manage business service property records.

**business service property business service group**

A classification for business service property at the business service level. This is generally a business service name. A business service level contains one or more business service property groups. Each business service property group may contain zero or more business service property records.

**business service property key**

A unique name that identifies the business service property globally in the system.

**business service property utilities**

A utility API used in business service development to access EnterpriseOne business service property data.

**business service property value**

A value for a business service property.

**business service repository**

A source management system, for example ClearCase, where business service artifacts and build files are stored. Or, a physical directory in network.

**business services server**

The physical machine where the business services are located. Business services are run on an application server instance.

**business services source file or business service class**

One type of business service artifact. A text file with the .java file type written to be compiled by a Java compiler.

**business service value object template**

The structural representation of a business service value object used in a C-business function.

**Business Service Value Object Template Utility**

A utility used to create a business service value object template from a business service value object.

**business services server artifact**

The object to be deployed to the business services server.

**business view**

A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.

**central objects merge**

A process that blends a customer's modifications to the objects in a current release with objects in a new release.

**central server**

A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central

server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.

**charts**

Tables of information in JD Edwards EnterpriseOne that appear on forms in the software.

**check-in repository**

A repository for developers to check in and check out business service artifacts. There are multiple check-in repositories. Each can be used for a different purpose (for example, development, production, testing, and so on).

**checksum**

A fixed-size datum computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage. JD Edwards EnterpriseOne uses the checksum to verify the integrity of packages that have been downloaded by recomputing the checksum of the downloaded package and comparing it with the checksum of the original package. The procedure that yields the checksum from the data is called a checksum function or checksum algorithm. JD Edwards EnterpriseOne uses the MD5 and STA-1 checksum algorithms.

**connector**

Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.

**Control Table Workbench**

An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.

**control tables merge**

A process that blends a customer's modifications to the control tables with the data that accompanies a new release.

**correlation data**

The data used to tie HTTP responses with requests that consist of business service name and method.

**credentials**

A valid set of JD Edwards EnterpriseOne username/password/environment/role, EnterpriseOne session, or EnterpriseOne token.

**cross-reference utility services**

Utility services installed in a BPEL/ESB environment that access EnterpriseOne cross-reference data.

**database credentials**

A valid database username/password.

**database server**

A server in a local area network that maintains a database and performs searches for client computers.

**Data Source Workbench**

An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

**deployment artifacts**

Artifacts that are needed for the deployment process, such as servers, ports, and such.

**deployment server**

A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

**direct connect**

A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one and store-and-forward.

**Do Not Translate (DNT)**

A type of data source that must exist on the iSeries because of BLOB restrictions.

**embedded application server instance**

An OC4J instance started by and running wholly within JDeveloper.

**edit code**

A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

**edit mode**

A condition of a form that enables users to change data.

**edit rule**

A method used for formatting and validating user entries against a predefined rule or set of rules.

**Electronic Data Interchange (EDI)**

An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

**embedded event rule**

An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.



**Employee Work Center**

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

**enterprise server**

A server that contains the database and the logic for JD Edwards EnterpriseOne.

**Enterprise Service Bus (ESB)**

Middleware infrastructure products or technologies based on web services standards that enable a service-oriented architecture using an event-driven and XML-based messaging framework (the bus).

**EnterpriseOne administrator**

An actor responsible for the EnterpriseOne administration system.

**EnterpriseOne credentials**

A user ID, password, environment, and role used to validate a user of EnterpriseOne.

**EnterpriseOne development client**

Historically called “fat client,” a collection of installed EnterpriseOne components required to develop EnterpriseOne artifacts, including the Microsoft Windows client and design tools.

**EnterpriseOne extension**

A JDeveloper component (plug-in) specific to EnterpriseOne. A JDeveloper wizard is a specific example of an extension.

**EnterpriseOne object**

A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

**EnterpriseOne process**

A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

**EnterpriseOne resource**

Any EnterpriseOne table, metadata, business function, dictionary information, or other information restricted to authorized users.

**Environment Workbench**

An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

**escalation monitor**

A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

**event rule**

A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

**explicit transaction**

Transaction used by a business service developer to explicitly control the type (auto or manual) and the scope of transaction boundaries within a business service.

**exposed method or value object**

Published business service source files or parts of published business service source files that are part of the published interface. These are part of the contract with the customer.

**fast path**

A command prompt that enables the user to move quickly among menus and applications by using specific commands.

**file server**

A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

**final mode**

The report processing mode of a processing mode of a program that updates or creates data records.

**foundation**

A framework that must be accessible for execution of business services at runtime. This includes, but is not limited to, the Java Connector and JDBj.

**FTP server**

A server that responds to requests for files via file transfer protocol.

**HTTP Adapter**

A generic set of services that are used to do the basic HTTP operations, such as GET, POST, PUT, DELETE, TRACE, HEAD, and OPTIONS with the provided URL.

**instantiate**

A Java term meaning “to create.” When a class is instantiated, a new instance is created.

**integration developer**

The user of the system who develops, runs, and debugs the EnterpriseOne business services. The integration developer uses the EnterpriseOne business services to develop these components.

**integration point (IP)**

The business logic in previous implementations of EnterpriseOne that exposes a document level interface. This type of logic used to be called XBP. In EnterpriseOne 8.11, IPs are implemented in Web Services Gateway powered by webMethods.

**integration server**

A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

**integrity test**

A process used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

**interface table**

See Z table.

**internal method or value object**

Business service source files or parts of business service source files that are not part of the published interface. These could be private or protected methods. These could be value objects not used in published methods.

**interoperability model**

A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

**in-your-face error**

In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

**jargon**

An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

**Java application server**

A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

**JDBNET**

A database driver that enables heterogeneous servers to access each other's data.

**JDEBASE Database Middleware**

A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

**JDECallObject**

An API used by business functions to invoke other business functions.

**jde.ini**

A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

**JDEIPC**

Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

**jde.log**

The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

**JDENET**

A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

**JDeveloper Project**

An artifact that JDeveloper uses to categorize and compile source files.

**JDeveloper Workspace**

An artifact that JDeveloper uses to organize project files. It contains one or more project files.

**JMS Queue**

A Java Messaging service queue used for point-to-point messaging.

**listener service**

A listener that listens for XML messages over HTTP.

**local repository**

A developer's local development environment that is used to store business service artifacts.

**Location Workbench**

An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

**logic server**

A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

**MailMerge Workbench**

An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

**Manual Commit transaction**

A database connection where all database operations delay writing to the database until a call to commit is made.

**master business function (MBF)**

An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

**master table**

See published table.

**media storage object**

Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

**message center**

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

**messaging adapter**

An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

**messaging server**

A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

**Monitoring Application**

An EnterpriseOne tool provided for an administrator to get statistical information for various EnterpriseOne servers, reset statistics, and set notifications.

**named event rule (NER)**

Encapsulated, reusable business logic created using event rules, rather than C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

**Object Configuration Manager (OCM)**

In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

**Object Librarian**

A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as

production and development) and enables objects to be easily moved from one environment to another.

### **Object Librarian merge**

A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

### **Open Data Access (ODA)**

An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

### **Output Stream Access (OSA)**

An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

### **package**

JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

### **package build**

A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: “Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions.” The process of creating a package build is often referred to, as it is in this example, simply as “a package build.”

### **package location**

The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path\_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

### **Package Workbench**

An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

### **Pathcode Directory**

The specific portion of the file system on the EnterpriseOne development client where EnterpriseOne development artifacts are stored.

### **patterns**

General repeatable solutions to a commonly occurring problem in software design. For business service development, the focus is on the object relationships and interactions.

For orchestrations, the focus is on the integration patterns (for example, synchronous and asynchronous request/response, publish, notify, and receive/reply).

**print server**

The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

**pristine environment**

A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

**processing option**

A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

**production environment**

A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

**Production Published Business Services Web Service**

Published business services web service deployed to a production application server.

**program temporary fix (PTF)**

A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

**project**

In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

**promotion path**

The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, 11 equals new project pending review, 21 equals programming, 26 equals QA test/review, 28 equals QA test/review complete, 38 equals in production, 01 equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

**proxy server**

A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

**published business service**

EnterpriseOne service level logic and interface. A classification of a published business service indicating the intention to be exposed to external (non-EnterpriseOne) systems.

**published business service identification information**

Information about a published business service used to determine relevant authorization records. Published business services + method name, published business services, or \*ALL.

**published business service web service**

Published business services components packaged as J2EE Web Service (namely, a J2EE EAR file that contains business service classes, business service foundation, configuration files, and web service artifacts).

**published table**

Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

**publisher**

The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

**QBE**

An abbreviation for query by example. In JD Edwards EnterpriseOne, the QBE line is the top line on a detail area that is used for filtering data.

**real-time event**

A message triggered from EnterpriseOne application logic that is intended for external systems to consume.

**refresh**

A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level.

**replication server**

A server that is responsible for replicating central objects to client machines.

**rules**

Mandatory guidelines that are not enforced by tooling, but must be followed in order to accomplish the desired results and to meet specified standards.

**secure by default**

A security model that assumes that a user does not have permission to execute an object unless there is a specific record indicating such permissions.

**Secure Socket Layer (SSL)**

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

**selection**

Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.



**serialize**

The process of converting an object or data into a format for storage or transmission across a network connection link with the ability to reconstruct the original data or objects when needed.

**Server Workbench**

An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number data source. The application also updates the Server Plan detail record to reflect completion.

**SOA**

Abbreviation for Service Oriented Architecture.

**softcoding**

A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.

**source repository**

A repository for HTTP adapter and listener service development environment artifacts.

**Specification merge**

A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.

**specification**

A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.

**Specification Table Merge Workbench**

An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.

**SSL Certificate**

A special message signed by a certificate authority that contains the name of a user and that user's public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in the user's public key.

**store-and-forward**

The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.

**subscriber table**

Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.

**super class**

An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the super class of "Oak" and "Elm," for example.

**table access management (TAM)**

The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.

**Table Conversion Workbench**

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

**table conversion**

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

**table event rules**

Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.

**terminal server**

A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

**transaction processing (TP) monitor**

A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

**transaction processing method**

A method related to the management of a manual commit transaction boundary (for example, start, commit, rollback, and cancel).

**transaction set**

An electronic business transaction (electronic data interchange standard document) made up of segments.

**trigger**

One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

**triggering event**

A specific workflow event that requires special action or has defined consequences or resulting actions.

**user identification information**

User ID, role, or \*public.

**User Overrides merge**

Adds new user override records into a customer's user override table.

**value object**

A specific type of source file that holds input or output data, much like a data structure passes data. Value objects can be exposed (used in a published business service) or internal, and input or output. They are comprised of simple and complex elements and accessories to those elements.

**versioning a published business service**

Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

**Versions List merge**

The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

**visual assist**

Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

**vocabulary override**

An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

**web application server**

A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

**web server**

A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

**Web Service Description Language (WSDL)**

An XML format for describing network services.

**Web Service Inspection Language (WSIL)**

An XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made.

**web service softcoding record**

An XML document that contains values that are used to configure a web service proxy. This document identifies the endpoint and conditionally includes security information.

**web service softcoding template**

An XML document that provides the structure for a soft coded record.

**Where clause**

The portion of a database operation that specifies which records the database operation will affect.

**Windows terminal server**

A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

**wizard**

A type of JDeveloper extension used to walk the user through a series of steps.

**workbench**

A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

**workflow**

The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

**workgroup server**

A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

**XAPI events**

A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

**XML CallObject**

An interoperability capability that enables you to call business functions.

**XML Dispatch**

An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

**XML List**

An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

**XML Service**

An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

**XML Transaction**

An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

**XML Transaction Service (XTS)**

Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

**Z event**

A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

**Z table**

A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

**Z transaction**

Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.



---

---

# Index

## A

---

### Add Procedure

- Comment, 6-5, 7-17
- Description, 6-4, 7-17
- Name, 6-4, 7-16
- Procedure Release, 6-5, 7-17
- Status, 6-5, 7-17
- Template Name, 6-5

adding a change table configuration, 4-11

adding a software master definition, 4-8

### Additional Change Table Definitions

- Continue, 5-10
- Quit, 5-10

## B

---

### Batch Application Selection, 4-12

- Batch Application, 5-10
- Change Table Type, 5-10
- Version, 5-10

batch process, build step process, 4-28

build change tables, build step process, 4-27

build cross-reference, build step process, 4-26

build data dictionary, build step process, 4-26

build inhouse package database, build step  
process, 4-26

build package database, build step process, 4-28

build planner database, build step process, 4-26

build pristine database, build step process, 4-26

build standalone database, build step process, 4-27

build step processes, 4-25

- batch process, 4-28

- build change tables, 4-27

- build cross-reference, 4-26

- build data dictionary, 4-26

- build in-house package database, 4-26

- build package database, 4-28

- build planner database, 4-26

- build pristine database, 4-26

- build standalone database, 4-27

- build store & forward database, 4-26

- compress, 4-27

- copy CD directory files, 4-27

- create CD file structure, 4-25

- create INF, 4-27

- data cleanup, 4-28

- manual build step, 4-25

- master central objects, 4-27

- package build, 4-25

- pre-build package, 4-26

- pre-build pristine, 4-26

- undefined, 4-25

- virus check, 4-27

- Windows executable, 4-28

build steps

- list of step icons, 4-20

build store & forward database, build step

- process, 4-26

## C

---

### CD Configuration

- copying, 7-10

CD Configuration Director, 4-9, 8-10

CD information, 4-9

### CD Revisions

#### Build Steps

- Executable, 7-9, 8-13

- Sequence, 7-9, 8-13

- Status, 7-9, 8-13

- Target Folder, 7-9, 8-13

- Type, 7-9, 8-13

- Value/Parameter, 7-9, 8-13

#### CD Information

- Comment, 7-9, 8-13

- Image Path, 7-9, 8-13

- Status, 7-8, 8-12

#### Director Steps

- Package/Change Table Form, 7-10, 8-14

- Software Update Form, 7-10, 8-14

#### Directory Structure

- File Filter, 7-10, 8-14

- Full Path, 7-10, 8-14

- Source Path, 7-10, 8-14

CD Templates, 7-1

- adding, 7-4

- setting up, 7-2

- Template Director, 7-4

Change Table Configurations, 5-1

change table configurations

- adding, 4-11

- setting up, 4-11
- Change Table Definitions
  - adding, 5-5
  - Batch Application, 5-4
  - Change Table Type, 5-4
  - Configuration Status, 5-4
  - setting up, 5-2
  - Source Environment, 5-4
  - Source Release, 5-3
  - Target Release, 5-3
  - Version, 5-4
- Change Table Director, 4-11, 5-4, 5-5
- Additional Change Table Definitions, 5-10
- Batch Application Selection, 5-9
- Source Environment Selection, 5-7
- Target Release Selection, 5-6
- Change Table Revisions, 5-1
- change tables, configuring, 4-11
- compress, build step process, 4-27
- configuring change tables, 4-11
- Control Change Tables
  - viewing, 7-23
- Copy CD Configuration
  - Copy From, 7-11
  - Copy To, 7-12
- copy CD directory files, build step process, 4-27
- Copy Procedure
  - Copy From Procedure, 7-18
  - Copy To Procedure, 7-18
- copying a software master definition, 4-24
- create CD file structure, build step process, 4-25
- create INF, build step process, 4-27
- creating a data dictionary in the pristine data
  - source, 4-4
- creating a software master, 4-7
- creating the final software master, 4-9
- custom CD, 4-30

## D

---

- Data Cleanup Procedure
  - adding, 7-16
- Data Cleanup Procedure Application
  - using, 6-2
- Data Cleanup Procedure Templates, 7-15
  - build steps, 6-9
  - coping, 6-8, 6-9
  - Procedure Revision Steps, 6-7
- Data Cleanup Procedures
  - adding, 6-3
- data cleanup procedures
  - adding, 6-3
  - application, 6-1
- data cleanup, build step process, 4-28
- data sources
  - creating a data dictionary in the pristine data
    - source, 4-4
- defining a software master, 4-8
- deleting a software master, 4-19
- details of all build step processes, 4-25

## Index-2

- Distributed Development, 2-2
  - Localizations, 2-2
  - Multiple Physical Locations, 2-2
  - Multiple Platforms, 2-2
  - Outsourcing and Contractors, 2-2

## F

---

- forms
  - Add Procedure, 7-15
  - Additional Change Table Definitions, 5-5
  - Batch Application Selection, 4-12
  - CD Configuration Director, 4-9, 8-9
  - CD Information, 8-9
  - CD information, 4-9
  - CD Revisions, 4-20, 4-31, 7-1, 7-2, 8-9
  - Change Table Director, 4-11, 5-4
  - Copy CD Configuration, 4-20, 4-31, 7-2
  - Copy Change Table Definitions, 5-2
  - Copy Procedure, 6-9, 7-15
  - Language Group Information, 8-2
  - Language Information, 8-3
  - OMW Project Search & Select, 8-3
  - Procedure Definition Director, 6-3, 7-15
  - Procedure Revision, 6-10, 7-15
  - Procedure Revision Information, 6-3
  - Report Output Destination, 7-13
  - Select Languages, 8-3
  - Software Update Control Tables, 7-24
  - Source Batch Application Selection, 5-5
  - Source Environment Selection, 4-11, 5-4
  - Target Release Selection, 4-11, 5-4
  - Template Director, 4-31, 7-2
  - Template Information, 7-2
  - Work With CD Templates, 4-31
  - Work with CD Templates, 7-1
  - Work With Change Table Definitions, 5-1, 5-4
  - Work With Index Change Table, 7-21
  - Work with Language Groups, 8-2, 8-3, 8-9
  - Work With OMW Project Detail, 8-3
  - Work with OMW Project Detail, 8-3
  - Work With Procedure, 6-9, 6-10
  - Work with Procedure, 6-1, 6-3, 7-15
  - Work With Software Mastering, 4-19, 4-20, 7-13, 8-9
  - Work With Table Change Table, 7-18

## I

---

- icons, for build steps, 4-20
- Index Change Tables, 7-21
  - viewing, 7-20

## L

---

- Language Group Information, 8-4
  - All, 8-4
  - Data File Path, 8-4
  - Description, 8-4
  - Language Group, 8-4
  - Project, 8-4



## M

---

manual build step, build step process, 4-25  
master central objects, build step process, 4-27

## O

---

OMW Project Search & Select, 8-4, 8-5, 8-6, 8-7, 8-8  
Target Release Number, 8-8

## P

---

package build, build step process, 4-25  
Package Updates, 8-8  
    creating, 8-8  
path tokens, 7-12  
    relative  
        \$CLIENT, 7-12  
        \$DEP, 7-12  
        \$IMAGE, 7-12  
        \$TEMP, 7-12  
pre-build package, build step process, 4-26  
pre-build pristine, build step process, 4-26  
pristine data source, creating a data dictionary, 4-4  
Procedure Definition Director, 6-3, 7-16  
    Add Procedure, 7-16  
    Copy From Procedure, 6-9  
    Copy To Procedure, 6-9  
Procedure Director  
    Add Procedure, 6-3  
    Comment, 6-7  
    Description, 6-6  
    Name, 6-6  
    Procedure Revision Information, 6-5  
    Release, 6-7  
    Status, 6-7  
Procedure Revision, 7-17  
    Comment, 6-10, 7-18  
    Description, 6-10, 7-17  
    Name, 6-10, 7-17  
    Release, 6-10, 7-18  
    Status, 6-10, 7-17  
Product Packaging  
    implementation, 1-1  
proofing a software master definition, 4-24

## R

---

relative and absolute paths  
    converting, 7-13  
relative path tokens, 7-13

## S

---

Select Languages, 8-8  
    Data Source, 8-8  
    L, 8-8  
    Language, 8-8  
    Media Objects Data Source, 8-8  
setting up a software master definition, 4-8  
setting up change table configurations, 4-11

Software Change Management  
    Build Management, 2-1  
    Configuration Management, 2-1  
    Distributed Code Management, 2-2  
    Problem Tracking, 2-1  
    Project Management, 2-2  
    Promotion Management, 2-1  
    Release Management, 2-1  
    Version Control, 2-1  
software change management, 2-1  
software master  
    creating, 4-7  
    creating the final, 4-9  
    defining, 4-8  
    deleting, 4-19  
    updating JD Edwards EnterpriseOne, 4-16  
software master checklist, creating, 4-1  
software master definition, 4-19  
    adding, 4-8  
    copying, 4-24  
    proofing, 4-24  
    revising, 4-19  
    setting up, 4-8  
    validating, 4-13  
software master templates, 4-25  
    build step processes, 4-25  
    custom CD, 4-30  
    update CD template, 4-28  
Software Mastering Tools, 2-3  
    Change Table Configuration Director, 2-3  
    Data Cleanup Procedure Application, 2-4  
    Software Mastering Director, 2-3  
Software Update Control Tables, 7-24  
    From Release, 7-25  
    To Release, 7-25  
Source Environment Selection, 4-11  
    Description, 5-8  
    Source Environment, 5-8  
    Source Release, 5-9

## T

---

Table Change Table, 7-18  
    viewing, 7-18  
Target Release Selection, 4-11  
    CD Name, 5-7  
    Description, 5-7  
    Release, 5-7  
    Target Release, 5-7  
Template Director  
    CD Revisions, 7-7  
    Template Information, 7-5  
Template Information  
    Build Phase, 7-7  
    Description, 7-7  
    Release, 7-7  
    Type, 7-7  
Template Options, 8-2  
    Data Source, 8-2  
    Release, 8-2

- Translation Update Templates
  - creating, 8-1
  - processing options, 8-2
- Translation Updates, 8-1
  - collecting, 8-2
- Translation Updates Database
  - creating, 8-3

## U

---

- undefined, build step process, 4-25
- update CD template, 4-28
- using the software master to update JD Edwards EnterpriseOne, 4-16

## V

---

- validating a software master definition, 4-13
- virus check, build step process, 4-27

## W

---

- Windows executable, build step process, 4-28
- Work With CD Templates
  - Build Phase, 7-4
  - Build Phase Description, 7-4
  - CD Name, 7-4
  - CD Type, 7-4
  - Comment, 7-4
  - Definitions Date, 7-4
  - Description, 7-4
  - One Off CD, 7-4
  - Release, 7-3
  - Type Description, 7-4
- Work With Index Change Table
  - Date Updated, 7-23
  - EnterpriseOne Only (Y/N), 7-23
  - From Release, 7-23
  - Index Change Type, 7-23
  - Index Identifier, 7-23
  - Program ID, 7-23
  - Table Name, 7-22
  - Time Updated, 7-23
  - To Release, 7-23
  - User ID, 7-23
  - Work Stn ID, 7-23
- Work With Language Group
  - Database File Path, 8-3
  - Description, 8-3
  - Language, 8-4
  - Language Group, 8-3
  - OMW Project Type, 8-4
- Work With Language Groups, 8-3
- Work with OMW Project Detail, 8-4
  - Description, 8-4
  - OMW Project Name, 8-4
- Work With Procedure
  - Comment, 6-2, 7-16
  - Description, 6-2, 7-16
  - Name, 6-2, 7-15
  - Release, 6-2, 7-15

- Status, 6-2, 7-16
- Type, 6-2, 7-16
- User ID, 6-3, 7-16
- Work With Software Mastering, 8-9, 8-10
  - CD Master Date, 8-10
  - Comment, 8-10
  - Definitions Date, 8-10
  - One Off CD, 8-10
  - Status Description, 8-10
- Work With Table Change Table
  - Date Updated, 7-20
  - EnterpriseOne Only (Y/N), 7-20
  - From Release, 7-20
  - Program ID, 7-20
  - Table Change Type, 7-20
  - Table Name, 7-19
  - Time Updated, 7-20
  - To Release, 7-20
  - User ID, 7-20
  - Work Stn ID, 7-20