ORACLE
JD EDWARDS ENTERPRISEONE

JD Edwards EnterpriseOne Tools 8.97 Product Packaging Guide

October 2007

ORACLE

JD Edwards EnterpriseOne Tools 8.97 Product Packaging Guide
SKU E1_TOOLSTPP-B1007

Contents

About This Documentation Preface

JD Edwards EnterpriseOne implementation guides provide you with the information that you need to implement and use JD Edwards EnterpriseOne applications from Oracle.

This preface discusses:

- JD Edwards EnterpriseOne application prerequisites.
- Application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common fields in implementation guides.

Note. Implementation guides document only elements, such as fields and check boxes, that require additional explanation. If an element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common fields for the section, chapter, implementation guide, or product line. Fields that are common to all JD Edwards EnterpriseOne applications are defined in this preface.

JD Edwards EnterpriseOne Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use JD Edwards EnterpriseOne applications.

You might also want to complete at least one introductory training course, if applicable.

You should be familiar with navigating the system and adding, updating, and deleting information by using JD Edwards EnterpriseOne menus, forms, or windows. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your JD Edwards EnterpriseOne applications most effectively.

Application Fundamentals

Each application implementation guide provides implementation and processing information for your JD Edwards EnterpriseOne applications.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals implementation guide. Most product lines have a version of the application fundamentals implementation guide. The preface of each implementation guide identifies the application fundamentals implementation guides that are associated with that implementation guide.

The application fundamentals implementation guide consists of important topics that apply to many or all JD Edwards EnterpriseOne applications. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals implementation guides. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Download documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on Oracle's PeopleSoft Customer Connection website. Through the Documentation section of Oracle's PeopleSoft Customer Connection, you can download files to add to your Implementation Guides Library. You'll find a variety of useful and timely materials, including updates to the full line of JD Edwards EnterpriseOne documentation that is delivered on your implementation guides CD-ROM.

Important! Before you upgrade, you must check Oracle's PeopleSoft Customer Connection for updates to the upgrade instructions. Oracle continually posts updates as the upgrade process is refined.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Downloading Documentation

In addition to the complete line of documentation that is delivered on your implementation guide CD-ROM, Oracle makes JD Edwards EnterpriseOne documentation available to you via Oracle's website. You can download PDF versions of JD Edwards EnterpriseOne documentation online via the Oracle Technology Network. Oracle makes these PDF files available online for each major release shortly after the software is shipped.

See Oracle Technology Network, http://www.oracle.com/technology/documentation/psftent.html.

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps

Resource	Navigation
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs)	Implement, Optimize + Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
Implementation guides support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in implementation guides:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and JD Edwards EnterpriseOne or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.
" " (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Visual Cues

Implementation guides contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the JD Edwards EnterpriseOne system.

Note. Example of a note.

If the note is preceded by *Important!,* the note is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Cross-References

Implementation guides provide cross-references either under the heading "See Also" or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: "(FRA) Hiring an Employee"

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in implementation guides:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in implementation guides:

- USF (U.S. Federal)

- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments and suggestions are important to us. We encourage you to send us your feedback about our PeopleBooks and other reference and training materials. Please include the release numbers for the PeopleTools and applications that you are currently using. Email your comments to PSOFT-INFODEV_US@ORACLE.COM.

Common Fields Used in Implementation Guides

Address Book Number Enter a unique number that identifies the master record for the entity. An address book number can be the identifier for a customer, supplier, company, employee, applicant, participant, tenant, location, and so on. Depending on the application, the field on the form might refer to the address book number as the customer number, supplier number, or company number, employee or applicant ID, participant number, and so on.

As If Currency Code Enter the three-character code to specify the currency that you want to use to view transaction amounts. This code enables you to view the transaction amounts as if they were entered in the specified currency rather than the foreign or domestic currency that was used when the transaction was originally entered.

Batch Number Displays a number that identifies a group of transactions to be processed by the system. On entry forms, you can assign the batch number or the system can assign it through the Next Numbers program (P0002).

Batch Date Enter the date in which a batch is created. If you leave this field blank, the system supplies the system date as the batch date.

Batch Status Displays a code from user-defined code (UDC) table 98/IC that indicates the posting status of a batch. Values are:

Blank: Batch is unposted and pending approval.

A: The batch is approved for posting, has no errors and is in balance, but has not yet been posted.

D: The batch posted successfully.

E: The batch is in error. You must correct the batch before it can post.

P: The system is in the process of posting the batch. The batch is unavailable until the posting process is complete. If errors occur during the post, the batch status changes to *E*.

U: The batch is temporarily unavailable because someone is working with it, or the batch appears to be in use because a power failure occurred while the batch was open.

Branch/Plant Enter a code that identifies a separate entity as a warehouse location, job, project, work center, branch, or plant in which distribution and manufacturing activities occur. In some systems, this is called a business unit.

Business Unit Enter the alphanumeric code that identifies a separate entity within a business for which you want to track costs. In some systems, this is called a branch/plant.

Category Code Enter the code that represents a specific category code. Category codes are user-defined codes that you customize to handle the tracking and reporting requirements of your organization.

Company Enter a code that identifies a specific organization, fund, or other reporting entity. The company code must already exist in the F0010 table and must identify a reporting entity that has a complete balance sheet.

Currency Code Enter the three-character code that represents the currency of the transaction. JD Edwards EnterpriseOne provides currency codes that are recognized by the International Organization for Standardization (ISO). The system stores currency codes in the F0013 table.

Document Company Enter the company number associated with the document. This number, used in conjunction with the document number, document type, and general ledger date, uniquely identifies an original document.

If you assign next numbers by company and fiscal year, the system uses the document company to retrieve the correct next number for that company.

If two or more original documents have the same document number and document type, you can use the document company to display the document that you want.

Document Number Displays a number that identifies the original document, which can be a voucher, invoice, journal entry, or time sheet, and so on. On entry forms, you can assign the original document number or the system can assign it through the Next Numbers program.

Document Type Enter the two-character UDC, from UDC table 00/DT, that identifies the origin and purpose of the transaction, such as a voucher, invoice, journal entry, or time sheet. JD Edwards EnterpriseOne reserves these prefixes for the document types indicated:

P: Accounts payable documents.

R: Accounts receivable documents.

T: Time and pay documents.

I: Inventory documents.

O: Purchase order documents.

S: Sales order documents.

Effective Date Enter the date on which an address, item, transaction, or record becomes active. The meaning of this field differs, depending on the program. For example, the effective date can represent any of these dates:

- The date on which a change of address becomes effective.
- The date on which a lease becomes effective.
- The date on which a price becomes effective.
- The date on which the currency exchange rate becomes effective.
- The date on which a tax rate becomes effective.

Fiscal Period and **Fiscal Year**

Enter a number that identifies the general ledger period and year. For many programs, you can leave these fields blank to use the current fiscal period and year defined in the Company Names & Number program (P0010).

G/L Date (general ledger date)

Enter the date that identifies the financial period to which a transaction will be posted. The system compares the date that you enter on the transaction to the fiscal date pattern assigned to the company to retrieve the appropriate fiscal period number and year, as well as to perform date validations.

JD Edwards EnterpriseOne Tools Product Packaging Preface

This preface discusses JD Edwards EnterpriseOne Product Packaging companion documentation.

JD Edwards EnterpriseOne Product Packaging Companion Documentation

Additional, essential information describing the setup and design of Oracle's JD Edwards EnterpriseOne Tools resides in companion documentation. The companion documentation consists of important topics that apply to Oracle's JD Edwards EnterpriseOne Product Packaging as well as other JD Edwards EnterpriseOne Tools.

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 Customer Connection.

You should be familiar with the contents of these companion guides:

- JD Edwards EnterpriseOne Tools 8.97 Package Management Guide.
- JD Edwards EnterpriseOne Tools 8.97 Foundation Guide.
- JD Edwards EnterpriseOne Tools 8.97 Configurable Network Computing Implementation Guide.
- JD Edwards EnterpriseOne Tools 8.97 System Administration Guide.
- JD Edwards EnterpriseOne Tools 8.97 Server and Workstation Administration Guide.

See Also

JD Edwards EnterpriseOne Tools 8.97 Package Management Guide, "Getting Started with JD Edwards EnterpriseOne Package Management"

JD Edwards EnterpriseOne Tools 8.97 Foundation Guide, "Getting Started with Tools Foundation"

JD Edwards EnterpriseOne Tools 8.97 System Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools System Administration"

JD Edwards EnterpriseOne Tools 8.97 Server and Workstation Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools Server and Workstation Administration"

CHAPTER 1

Getting Started with JD Edwards EnterpriseOne Product Packaging

This chapter discusses:

- JD Edwards EnterpriseOne Product Packaging overview.
- JD Edwards EnterpriseOne Product Packaging implementation.

JD Edwards EnterpriseOne Product Packaging Overview

Oracle's JD Edwards EnterpriseOne Product Packaging Tools are used to manage the task of mastering the software. In this capacity, the tools oversee the entire mastering process from defining a software master to mastering the 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 a batch process to validate the software mastering information before mastering. Next, the tools provide a batch process to master the software. Finally, the tools provide a batch process to validate the final software master image.

This table defines some key terms and processes that are used with JD Edwards EnterpriseOne Product Packaging:

Software Mastering Director

This director enables administrators to build a mastered version of objects or modules. The director handles this functionality through a series of batch processes and manual tasks that build all of the components of a master. The final product is an image of the product, in a tree format, that administrators can burn onto a CD.

Change Table Configuration Director

This director enables administrators to define how to build the software master's change tables.

Distributed Development

This term 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 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.

CHAPTER 2

Understanding Product Packaging

This section provides an overview of JD Edwards EnterpriseOne Product Packaging.

Understanding Product Packaging

JD Edwards EnterpriseOne Product Packaging gives you the ability to create your own software mastering processes. You determine which objects should be shared between locations. You 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.

Software Change Management

Prompted by the move to distributed client and server environments, software change management has recently 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.

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.

Distributed Development

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.

CHAPTER 3

Working with Software Masters

This chapter provides an overview of software master creation and discusses how to:

- Prepare for software master creation.
- Create a software master.
- Delete a software master.
- Work with a software master definition.
- Work with software master templates.

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.

Preparing for Software Master Creation

This section discusses how to:

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

Setting Up the System TEMP Directory

This task creates a temporary directory that will be used to store the Oracle Supported Local Database (in MSDE) 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.

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. For each of the UBEs that are listed, search for the UBE, select the specified version, and change the path or filename to the specified value:

UBE	Version	Parameter	Value
R9640H	XJDE0006	Path of file to rename	$TEMP\Data \JDEMasters
R9640H	XJDE0007	Path of file to rename	$TEMP\Data \JDEMasters
R9640MF	XJDE0001	MSDE Database path	$TEMP\master
R9640E	XJDE0002	Component File Specifications	*.*

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
R9640MF	XJDE0001
R9640MG	XJDE0001
R9640MK	XJDE0001
R9640ML	XJDE0001
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.

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 PS812 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 - PS812. You perform this process only once.

To create pristine control tables:

1. On the Deployment Server, sign in to JD Edwards EnterpriseOne in the DEP812 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 - PS812.*

   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 - PS812.
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 - 812 data source.
11. For the PS812 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 - PS812 data source.
12. Activate the new mappings.
13. Quit Object Configuration Manager.

Verifying the Product Packaging Data Source

The JD Edwards EnterpriseOne Product Packaging process creates an MSDE database in the TEMP directory on the workstation on which the Product Package CD image is mastered. This database is used to store data that will be mastered on the CD. To copy tables into this database, you must verify that the OneWorld - Product Packaging data source and the Data Dictionary - PS812 data source are properly configured on the deployment server.

To verify the JD Edwards EnterpriseOne Product Packaging data source:

1. From the deployment server, sign in to JD Edwards EnterpriseOne in the DEP812 environment and enter *GH9011* in the Fast Path field.
2. Double-click Database Data Sources (P986115).
3. On Machine Search & Select, select the machine with the System - 812 data source.
4. On Work With Data Sources, click Find.
5. Select the OneWorld - Product Packaging data source.
6. On Data Source Revisions, verify that the OneWorld - Product Packaging data source is defined with these values:

   ODBC Data Source Name *OneWorld - Product Packaging*

   Data Source Type *N - MSDE/ODBC*

   Database Name *JDEMasters*
7. Click OK, and quit the Data Source application.

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 - PS812.*

   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 PS812 environment, add OCM mappings for F9200, F9202, F9203, F9207, F9210, F9211, and F9212 that point these tables to Data Dictionary - PS812.
14. Activate the new OCM mappings.

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.

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 - 812 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 - 812
F9755, F9757, F9759, F9760	Data Dictionary - 812

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.

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.

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.

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.

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.

After the mastering process is complete, the table should be in the JDEMasters database that is referenced by the OneWorld - Product Packaging data source that you set up earlier. In addition, records should exist in the table and index change table for the new custom table.

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 JDEMasters database.
2. Add a build step to copy the media object files from the deployment server to the image path directory.

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.

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.

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

Chapter 3, "Working with Software Masters," Working with Software Master Templates, page 33

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 DV812.
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 *E812UPDATE*. 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 *E812*. 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:\E812\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 Chapter 3, "Working with Software Masters," Setting the Package Build Application to the Mastering Mode, page 20.

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.

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 E812TAX 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 (*PS812*). 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 *E812*, 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.

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 DEP812 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.

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.

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.

Confirming the Software Master

After you create the software master image, you should visually check the directory structure and JDEMasters 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 JDEMasters 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 JDEMasters database and that they and their associated index change tables have records.

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.

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 an MSDE database (OWBAK_*packagename_pathcode*) and its corresponding files called OWBAK_*packagename_pathcode*.mdf and OWBAK_*packagename_pathcode*.ldf in the Planner\Package*Package_Name**PathCode*.bak\data directory. Only the tables and objects that are affected by the update will be backed up in the database.

    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.”

Report	Version	Description
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.

Report	Version	Description
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 DEP812 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.

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.

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.

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.

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.

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.

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.

Revising a Software Master Definition

Access the Work With Software Mastering form.

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.

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.

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 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*.

Task List

This table provides information for building the task list:

Task	Action
To run one or more steps:	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. 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.
To renumber the build steps:	From the Form menu, select Renumber. 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.
To add a subdirectory:	Click a directory for which you want to add a subdirectory and click Add. You can select any directory, including the Root directory. The software creates a new directory that is labeled Undefined and places it at the bottom of the structure.

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.

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.

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.

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.

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.

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.

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.

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.

Package Build

Type = 02

Performed Automatically = Yes

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

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.

Pre-Build Package

Type = 04

Performed Automatically = Yes

For Oracle internal use only.

Pre-Build Pristine

Type = 05

Performed Automatically = Yes

For Oracle internal use only.

Build Cross-Reference

Type = 06

Performed Automatically = Yes

For Oracle internal use only.

Build Data Dictionary

Type = 07

Performed Automatically = Yes

For Oracle internal use only.

Build In-House Package Database

Type = 08

Performed Automatically = Yes

For Oracle internal use only.

Build Planner Database

Type = 09

Performed Automatically = Yes

For Oracle internal use only.

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.

Build Store & Forward Database

Type = 11

Performed Automatically = Yes

For Oracle internal use only.

Build Standalone Database

Type = 12

Performed Automatically = Yes

For Oracle internal use only.

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.

Master Central Objects

Type = 14

Performed Automatically = No

For Oracle internal use only.

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.

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.

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.

Virus Check

Type = 18

Performed Automatically = Partial

For Oracle internal use only.

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.

Data Cleanup

Type = 20

Performed Automatically = No

For Oracle internal use only.

Build Package Database

Type = 21

Performed Automatically = Yes

For Oracle internal use only.

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.

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. This table shows the build steps and their sequence for an ASU CD:

Sequence Number	Build Step (Type)
10	Check Mastering Items (03 - Manual Build Step)
20	Cleanup Temporary Files (22 - Batch Process)
30	Cleanup Temp Master (22 - Batch Process)
40	Create CD File Structure (01 - Create the CD File Structure)
50	Build (02 - Package Build)
60	Create Database (22 - Batch Process)

Sequence Number	Build Step (Type)
70	Build Change Tables (13 - Build Change Tables)
80	Create/Load ASU Change Tables (22 - Batch Process)
90	Create/Load ASU Control Tables (22 - Batch Process)
100	Create/Load ASU Misc Tables (22 - Batch Process)
110	Create/Load New and Regen Tables (22 - Batch Process)
120	Create/Load ASU OL & VL Tables (22 - Batch Process)
130	ASU v/s Package Validation (22 - Batch Process)
140	Compact Database (22 - Batch Process)
150	Detach Database (22 - Batch Process)
160	Copy Database (22 - Batch Process)
170	Rename MSDE database files (22 - Batch Process)
180	Compress/Copy Database (15 - Compress)
190	Copy Package and Feature INFs (22 - Batch Process)
200	Copy CD Directory Files
210	Create INF
220	Create self-extracting executable
230	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 R9640L batch process using versions XJDE0001 through XJDE0005. This process deletes the previous software master from the build machine. This step deletes the contents of the $TEMP\data directory. It also drops the JDEMasters database, if present, by launching UBE R9640MK.

Sequence Number	Build Step (Type) with Additional Information
30	Cleanup Temp Master (22 - Batch Process) Launches the R9640L batch process, version XJDE0001 to delete the contents of $TEMP\Master folder.
40	Create CD File Structure (01 - Create CD Structure) Creates the CD file structure on the Image path.
50	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	Create Database (22 - Batch Process) Launches the R9640MF batch process using version XJDE0001. This process creates an empty MSDE database called JDEMasters in this directory: $TEMP/master.
70	Build Change Tables (13 - Build Change Tables) Builds the change tables based on the configuration that was set up during the director phase.
80	Create/Load ASU Change Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0003. This process creates change tables in the JDEMasters database, populating the tables from the Change Table data source.
90	Create/Load ASU Control Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0001. This process creates control tables in the JDEMasters database, and populates the tables from the Control Table data source.
100	Create/Load ASU Misc Tables (22 - Batch Process) Launches the R9670 batch process using version XJDE0002. This process creates miscellaneous tables in the JDEMasters 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.
110	Create/Load New & Regen Tables (22 - Batch Process) Launches the R96700 batch process using version XJDE0001. This process creates tables in the JDEMasters database that are marked as new or regenerate in the table change table or the table conversion scheduler.

Sequence Number	Build Step (Type) with Additional Information
120	Create/Load ASU OL & VL tables (22 - Batch Process) Launches the R9600400C batch process using version XJDE0001. This process copies to the JDEMasters 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.
130	ASU v/s Package Validation (22 - Batch Process) 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.
140	Compact Database (22 - Batch Process) Launches the R9640MG batch process using version XJDE0001, which runs the MSDE compaction tool to compact the JDEMasters MSDE database.
150	Detach Database (22 - Batch Process) Launches the R9640ML batch process using version XJDE0001, which detaches the JDEMasters MSDE database.
160	Copy Database (22 - Batch Process) Launches the R9640E batch process using version XJDE0002. This process copies the ASU database from $TEMP\master\JDEMasters.*df to $TEMP\data\JDEMasters.*df.
170	Rename MSDE Database files (22 - Batch Process) Launches the R9640H batch process using version XJDE0006 and XJDE007. This process renames the JDEMasters.*df files to *packagename*.*df files, where *packagename* is the name of the package associated with the database.
180	Compress/Copy Database (22 - Batch Process) Launches R9640E using version XJDE0003. This process compresses the ASU database (packagename.*df) files and copies them from the $TEMP/data directory to the final software master image path as Data.cab.

Sequence Number	Build Step (Type) with Additional Information
190	Copy Package and Feature INFs (22 - Batch Process) 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.
200	Copy CD Directory Files (17 - Copy CD Directory Files) Copies additional specified items into the image path. For example, all install programs are copied to the root of the image path.
210	Create INF (16 - Create INF) Creates the deployment INF file into the image path directory.
220	Create self-extracting exe (22 - Batch Process) 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.
230	Validate the Mastered CD (22 - Batch Process) Launches R9640D version XJDE0001, which performs post-mastering validation on the master that has been created. Refer to the report output for details.

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.

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

Chapter 3, “Working with Software Masters,” Revising a Software Master Definition, page 29

Chapter 3, “Working with Software Masters,” Loading a Subdirectory from an Enterprise Network Directory, page 31

Chapter 3, “Working with Software Masters,” Copying a Software Master Definition, page 32

Chapter 3, “Working with Software Masters,” Proofing a Software Master Definition, page 32

CHAPTER 4

Working with Change Table Configurations

This chapter provides an overview of change table revisions and discusses how to work with change table configurations.

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.

Working with Change Table Configurations

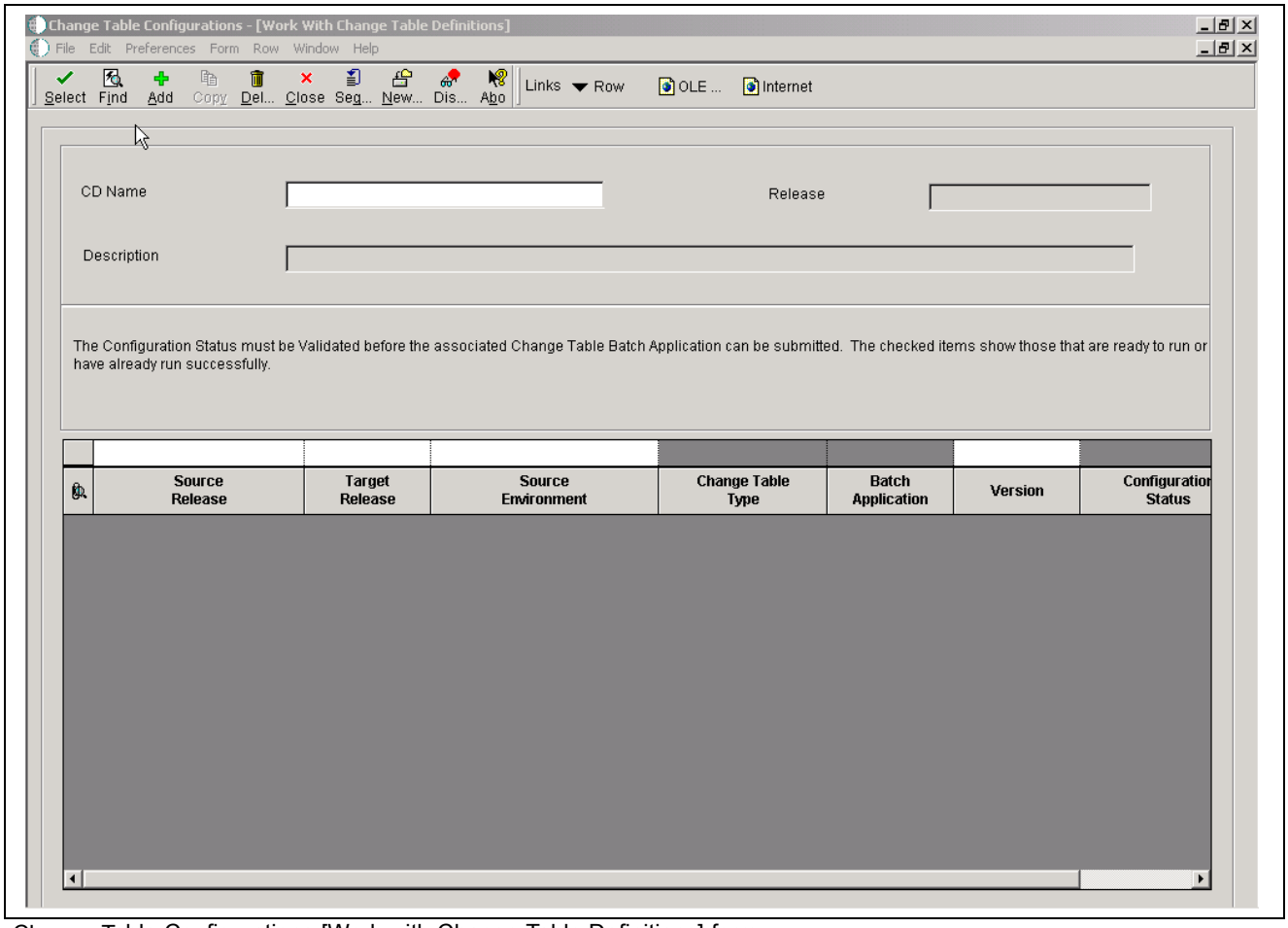
You copy change table configurations from one master definition to another master definition. This process copies all change table configurations associated with the master from which you are copying.

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.
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.

Setting Up Change Table Definitions

Access the Work With Change Table Definitions form.



Change Table Configurations [Work with Change Table Definitions] form

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.

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.

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.
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.

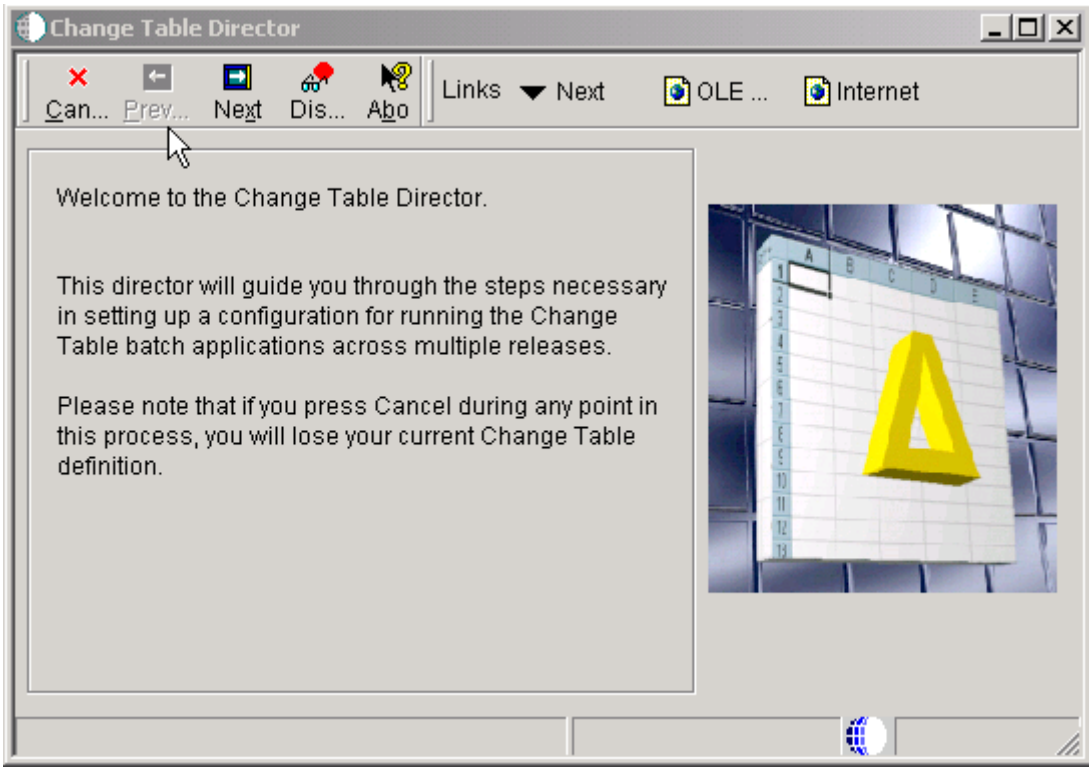
Adding a Change Table Definition

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

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.

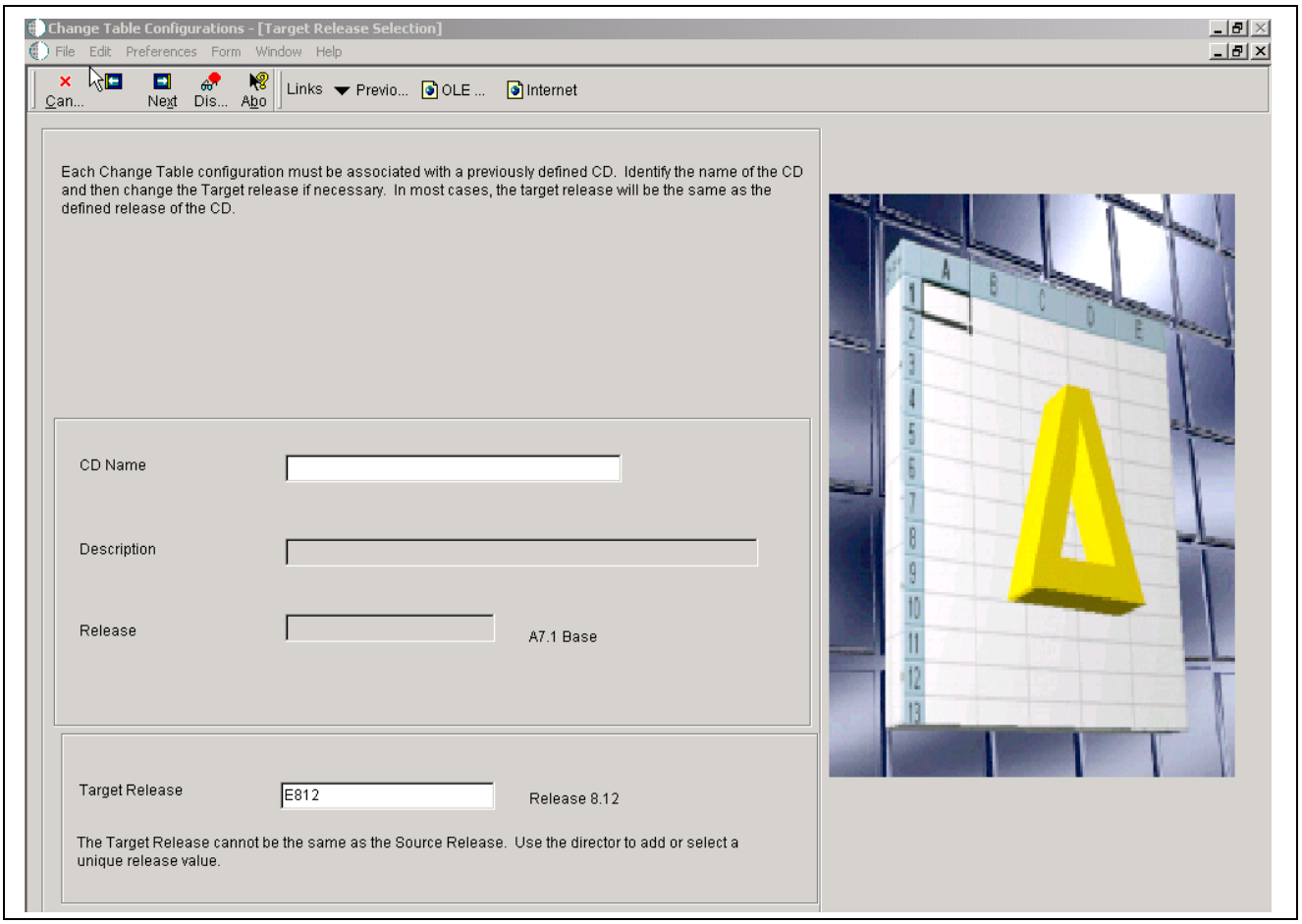


Change Table Director form

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.



Change Table Configurations [Target Release Selection] form

CD Name The unique name given to the software master.

Description A short description of the software master

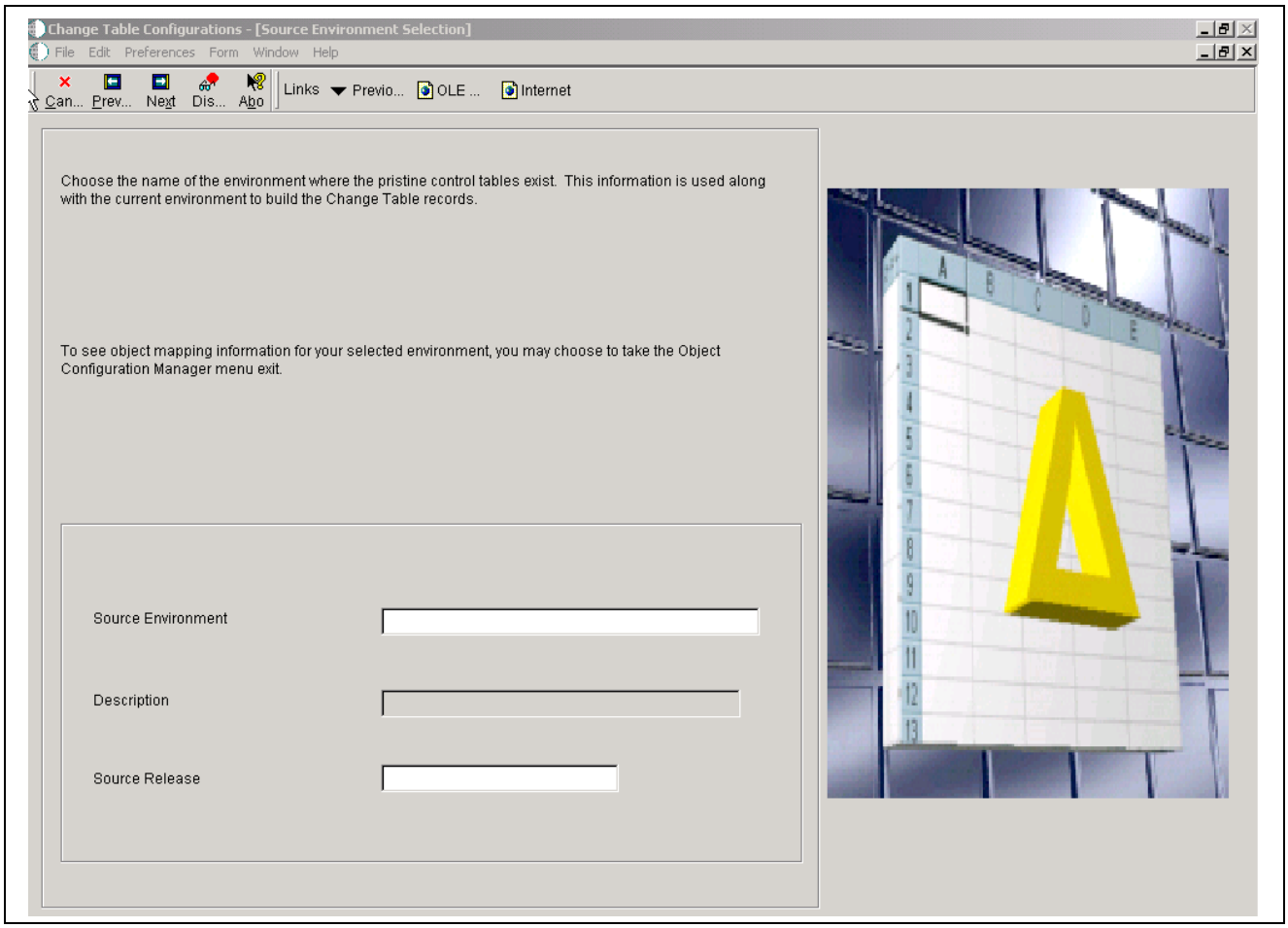
Release The release number defined in the software master.

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

Source Environment Selection

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

Access the Source Environment Selection form.



Change Table Configurations [Source Environment Selection] form

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

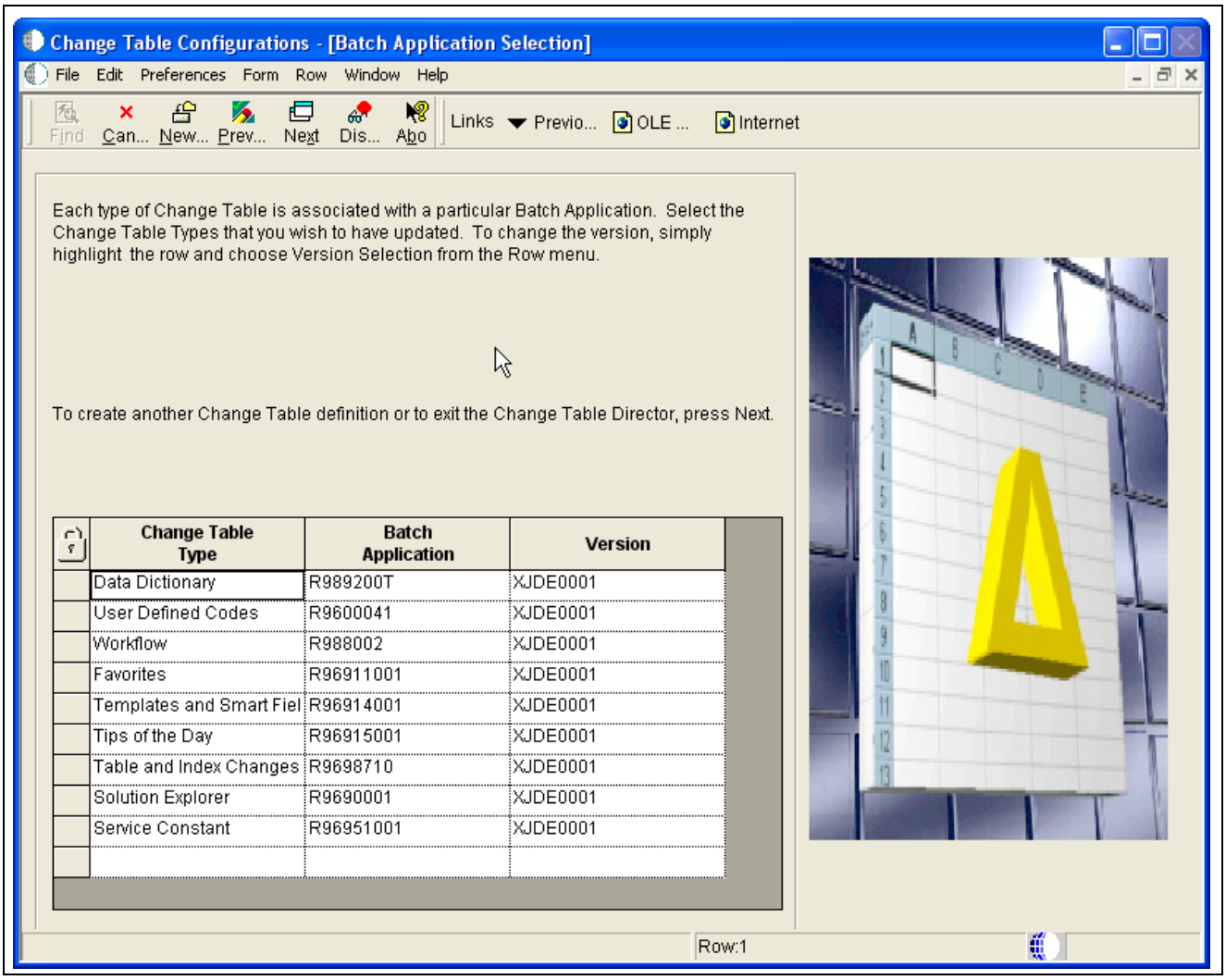
Description A user defined name or remark.

Source Release The release number defined in the release master.

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.



Change Table Configurations [Batch Application Selection] form

Change Table Type A user defined name or remark.

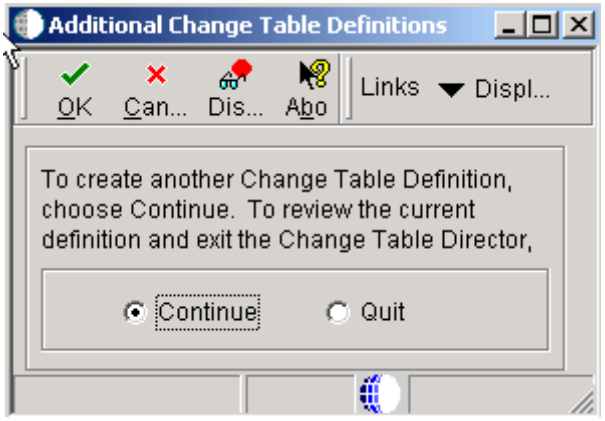
Batch Application Additional text that further describes or clarifies a field in the system.

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.

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.



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.

CHAPTER 5

Working with the Data Cleanup Procedure Application

This section provides an overview of Oracle's JD Edwards EnterpriseOne Data Cleanup Procedure application and discusses how to:

- Add a data cleanup procedure.
- Perform a data cleanup procedure and build step.

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.

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.

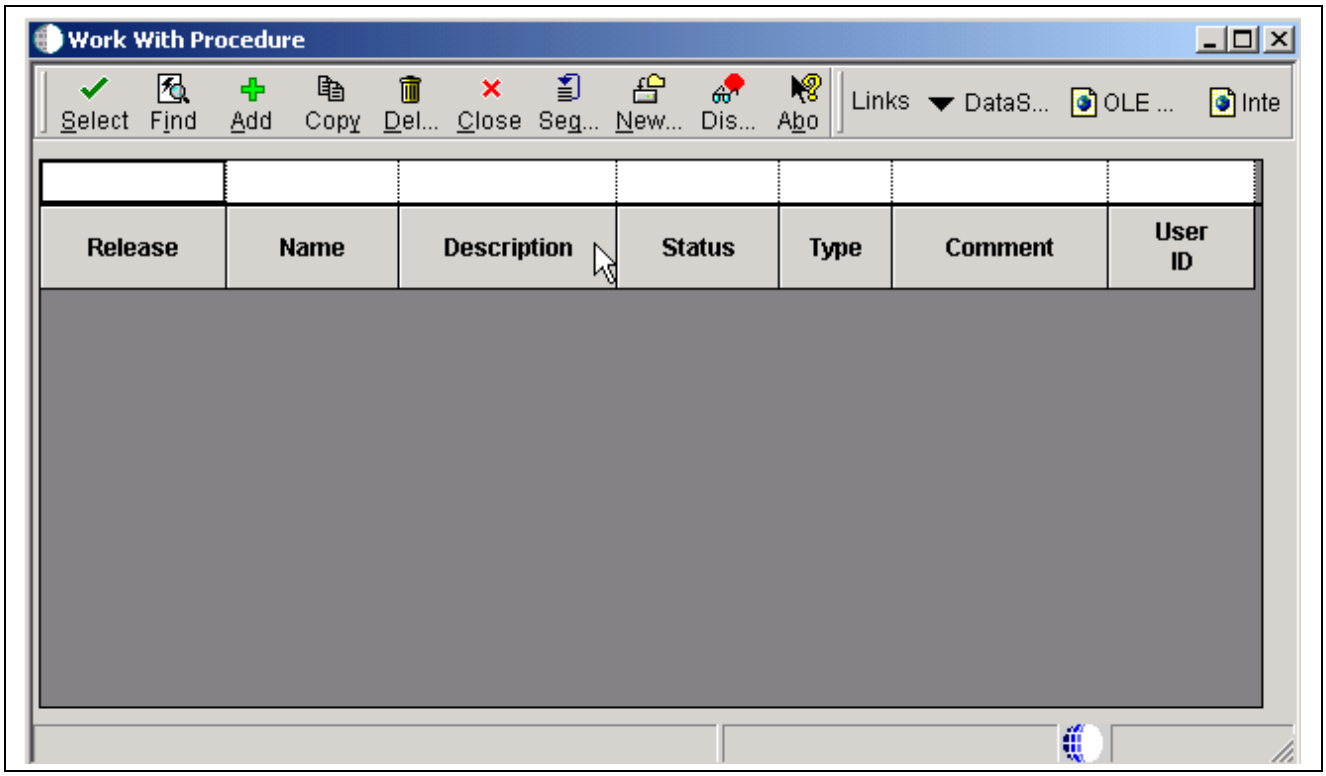
Forms Used to Work with Data Cleanup Procedures

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.

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.



Work with Procedure form

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.

Adding a Data Cleanup Procedure

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

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 it 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.

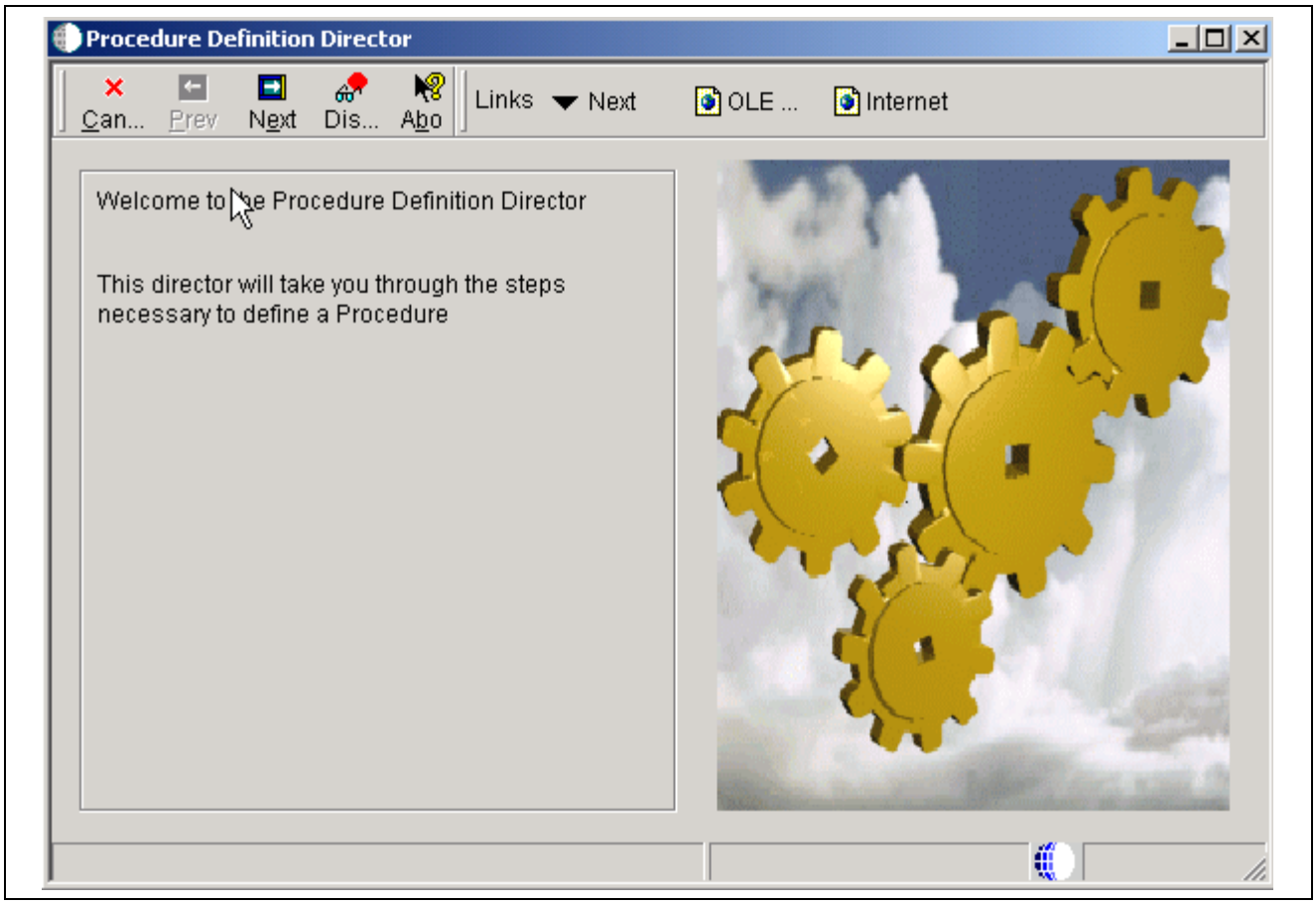
Adding Data Cleanup Procedures

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

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.

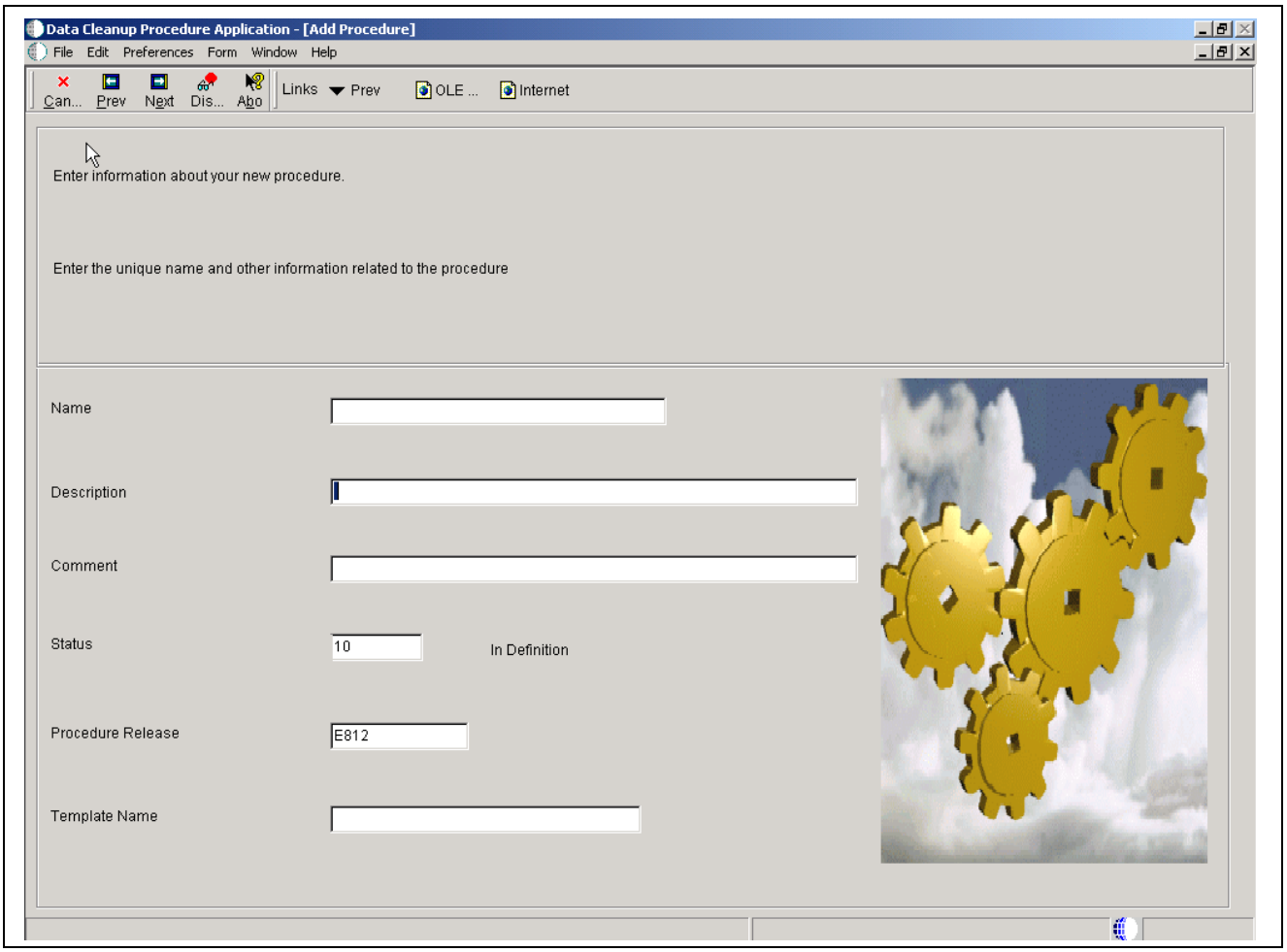


Procedure Definition Director form

Add Procedure

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

Access the Add Procedure form.



Data Cleanup Procedure Application [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 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

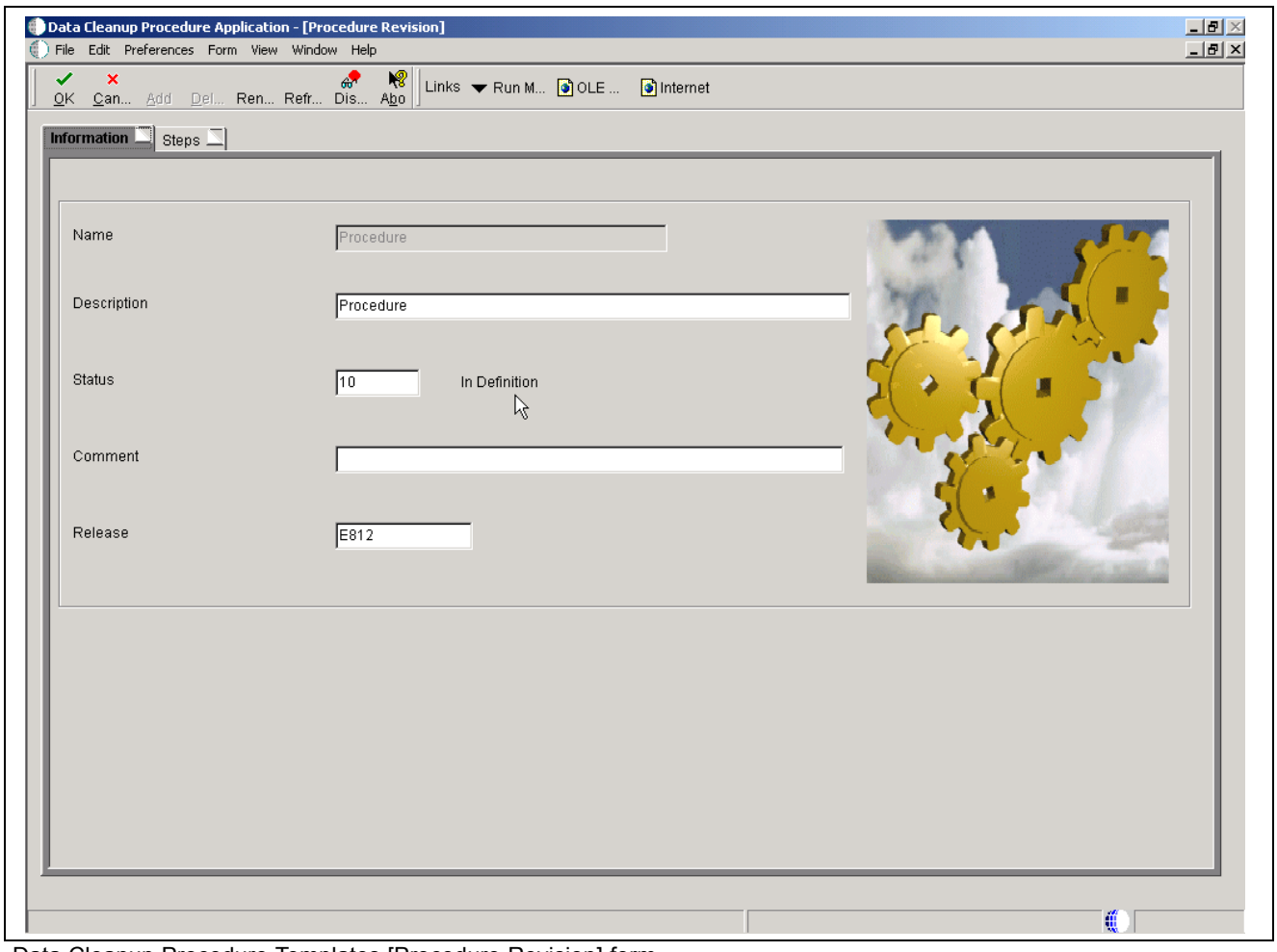
Procedure Release The system release number to which the data cleanup procedure applies.

Template Name The unique name given to a data cleanup procedure template.

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.



Data Cleanup Procedure Templates [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. 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

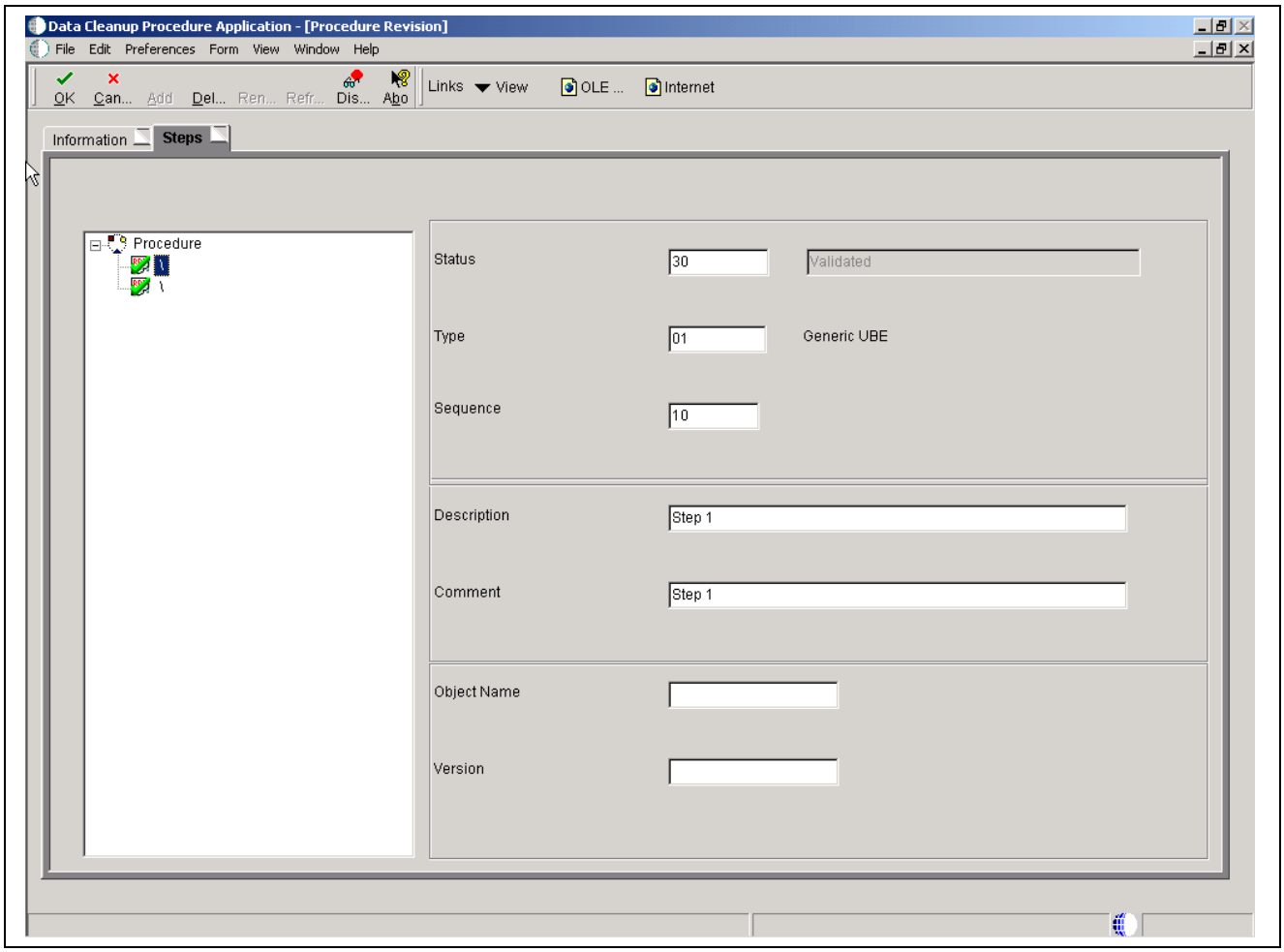
Comment A short informational comment that you provide about the data cleanup procedure.

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

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.



Data Cleanup Procedures Templates [Procedure Revision] form

Copying a Data Cleanup Procedure

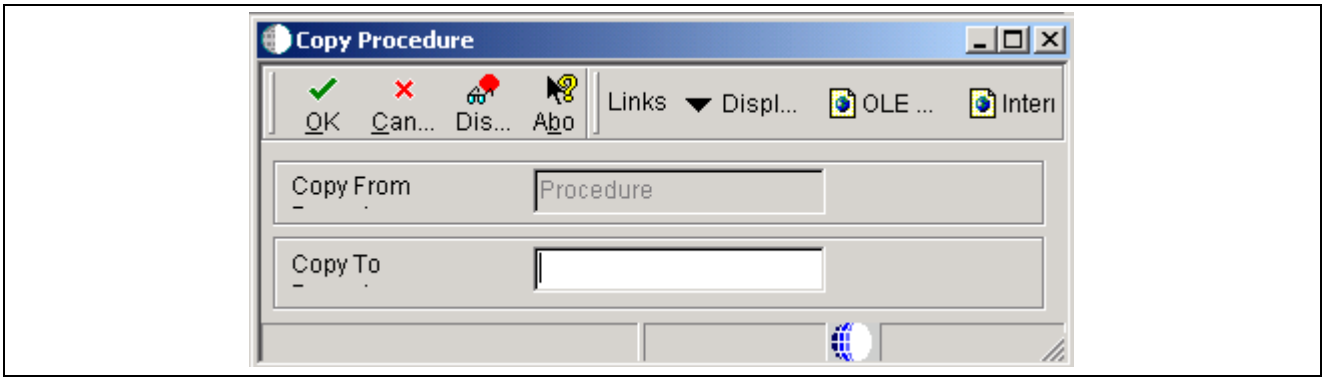
The JD Edwards EnterpriseOne Product Packaging Tool function is primarily used to create data cleanup procedures.

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.

Copying a Data Cleanup Procedure

Access the Procedure Definition Director form.



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.

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.

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.

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.

CHAPTER 6

Working with Advanced Operations

This chapter discusses how to:

- Work with CD templates.
- Convert relative and absolute paths.
- Work with Data Cleanup Procedure templates.
- View a table change table.
- View the index change tables.
- View the control change tables.

Working with CD Templates

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

- Work with CD Templates.
- Add a New Template

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.

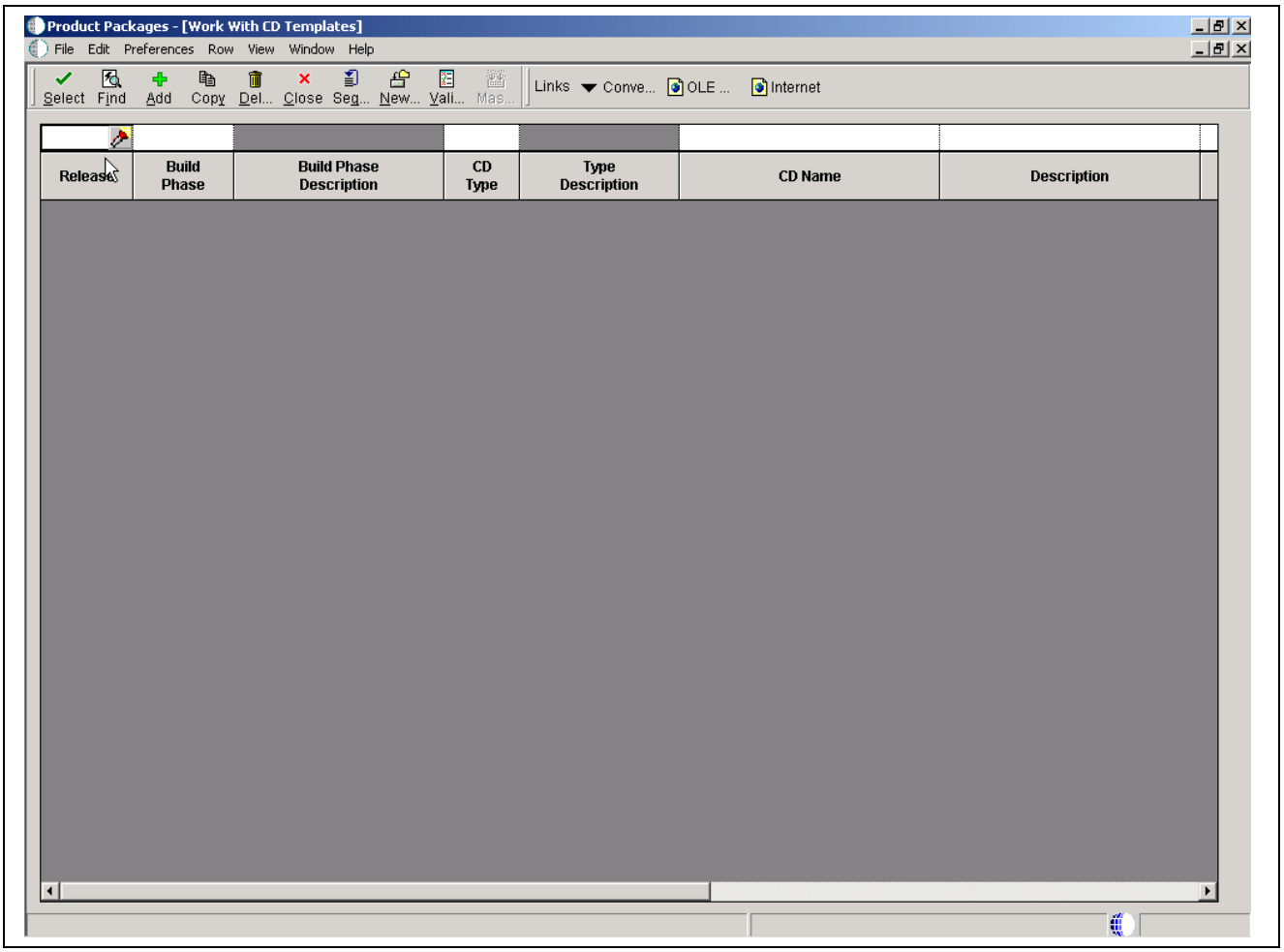
Forms Used with Mastering Templates

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.
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.

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.



Product Packages [Work with CD Templates] form

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.

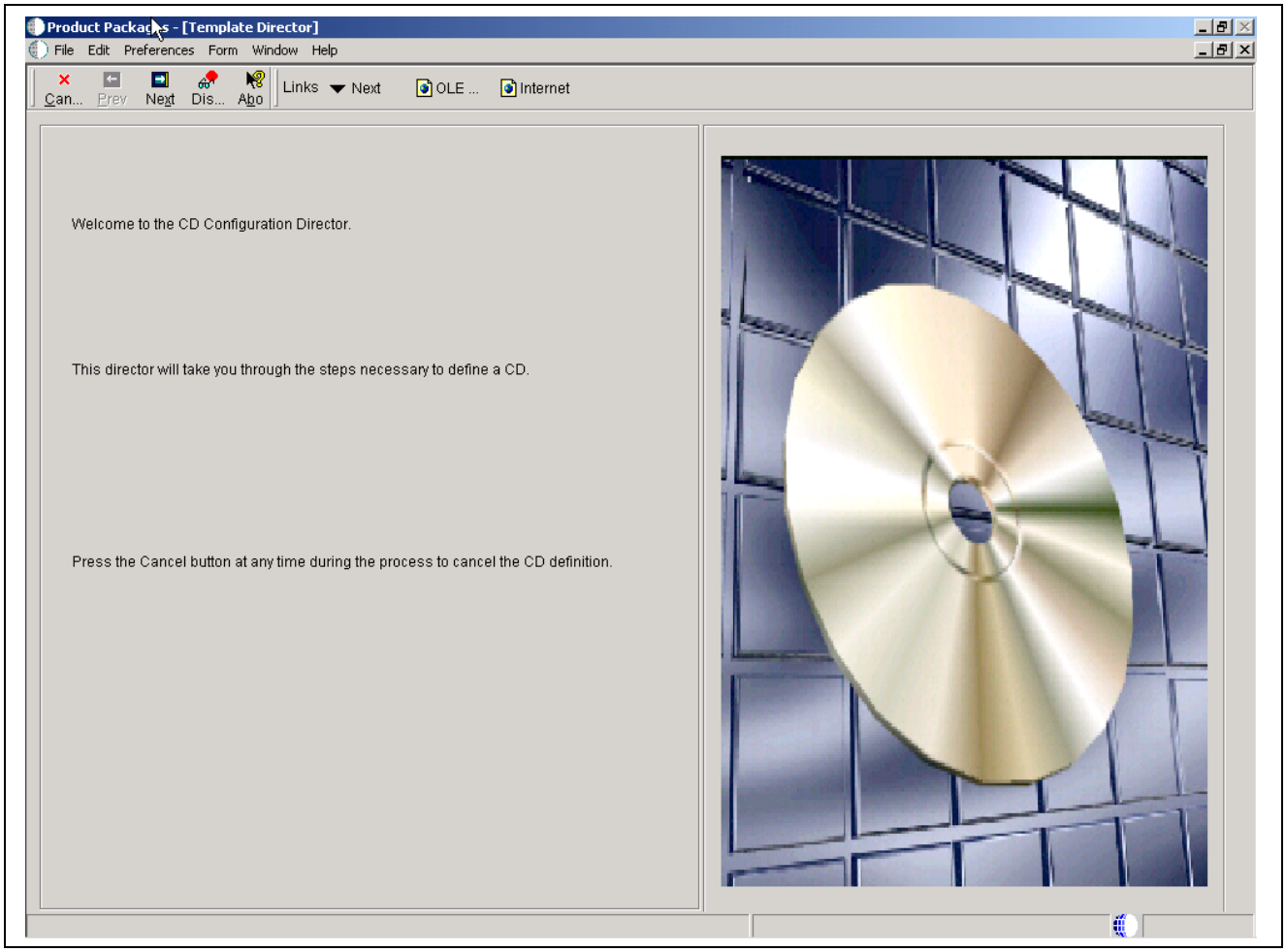
One Off CD Future use.

Adding a New CD Template

Access the Template Director form.

Template Director

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

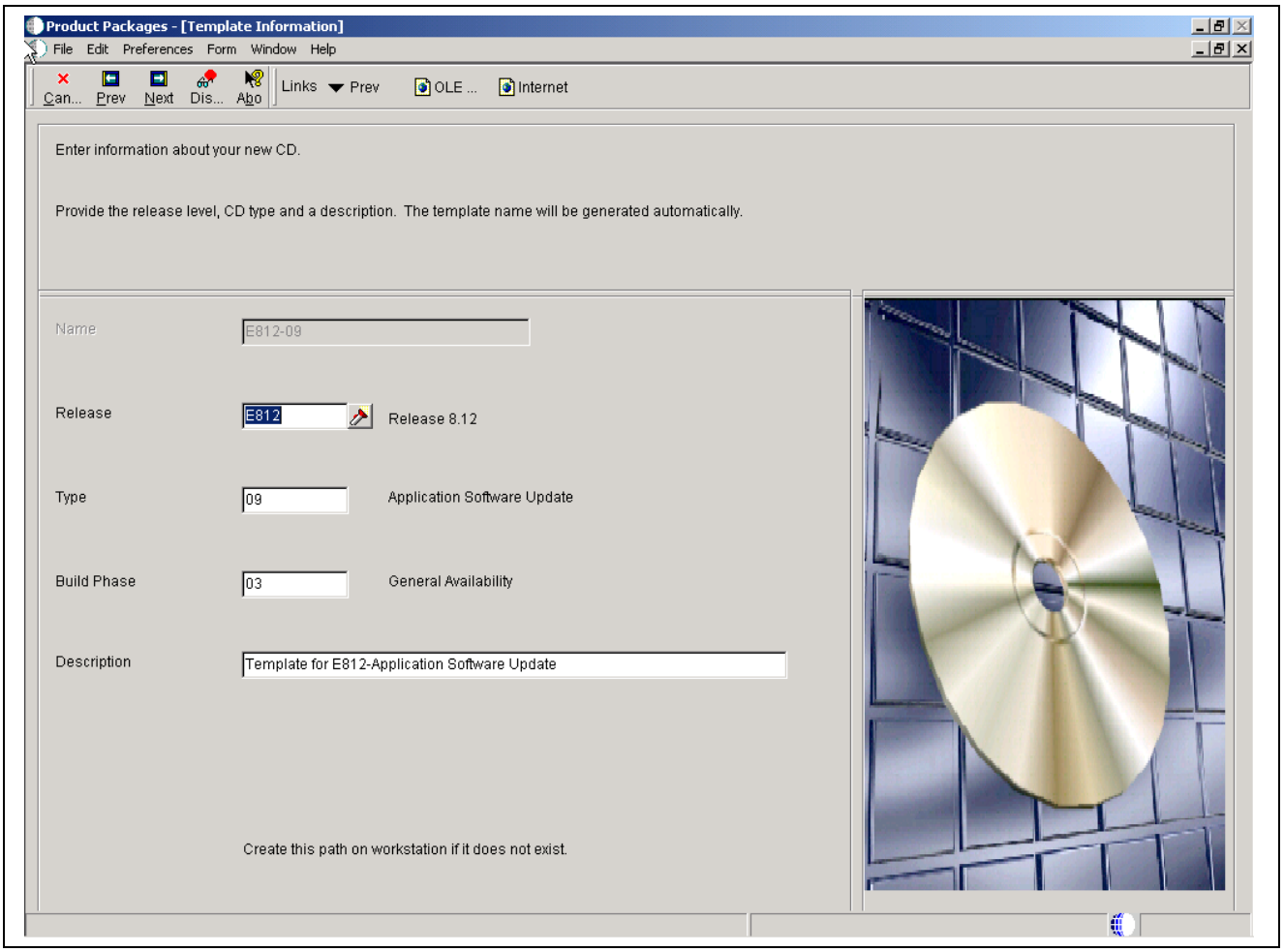


Product Packages [Template Director] form

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.



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.

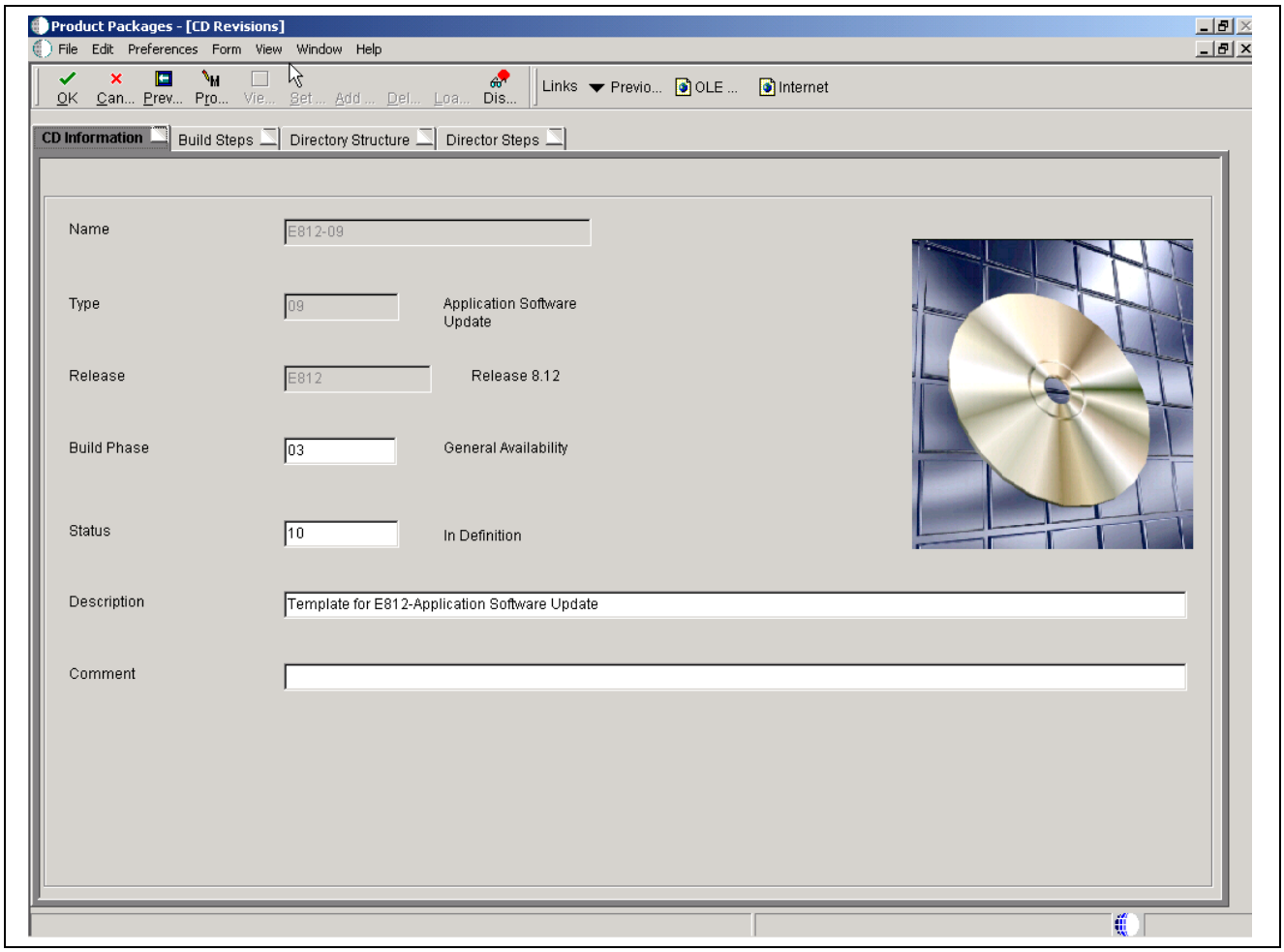
Build Phase This is the phase of development that your software master represents, such as an alpha, a beta, or a general availability phase. You can add or change the phases, since this is a user defined code. Use the visual-assist button to view a complete list of values for this field.

Description A user defined name or remark.

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.



Product Packages [CD Revisions] form

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 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.

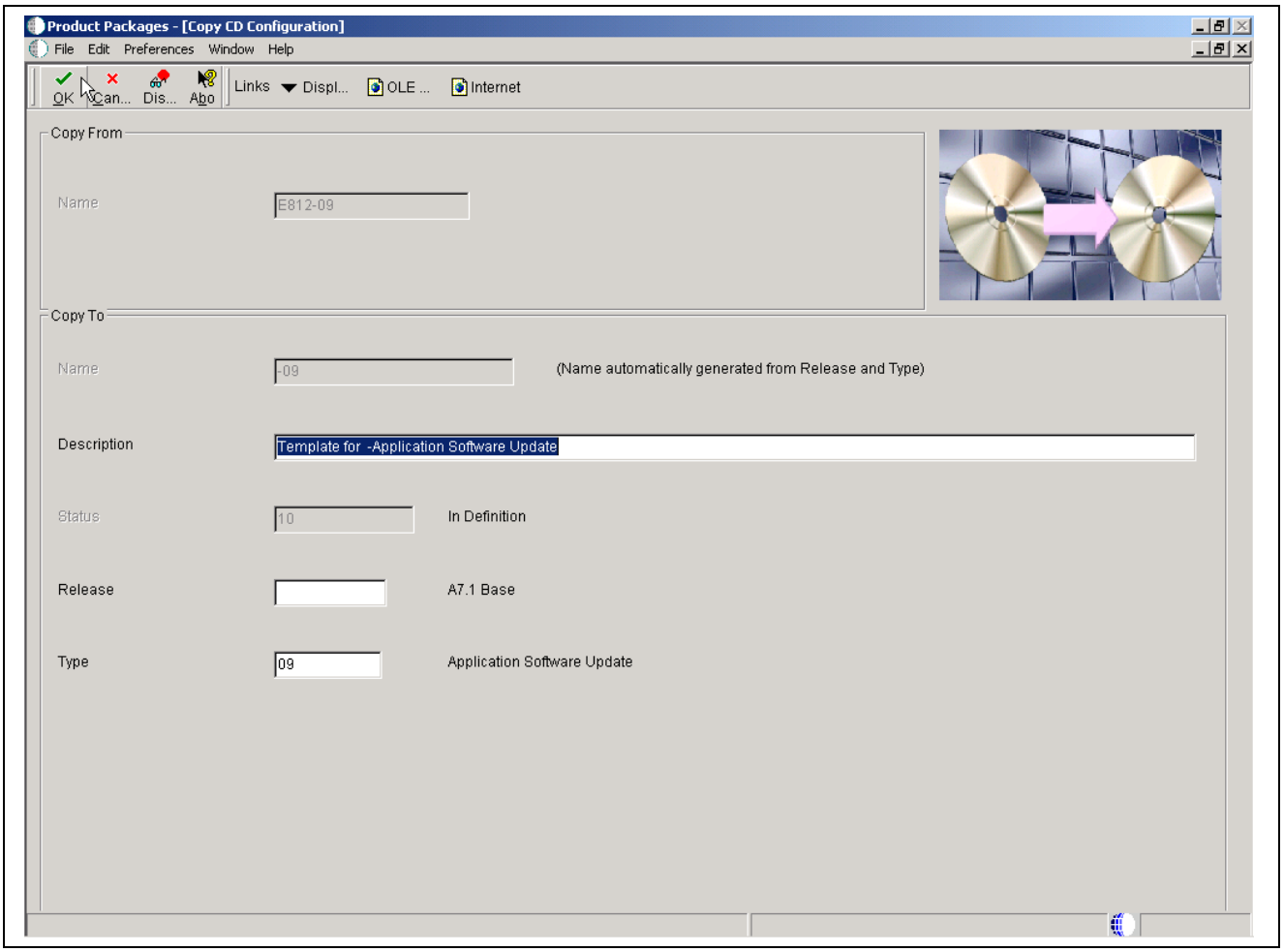
Package/Change Table Form Indicates the package form and change table form used.

Software Update Form Future use.

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.



Product Packages [Copy CD Configuration] form

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.

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.

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.

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.

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.

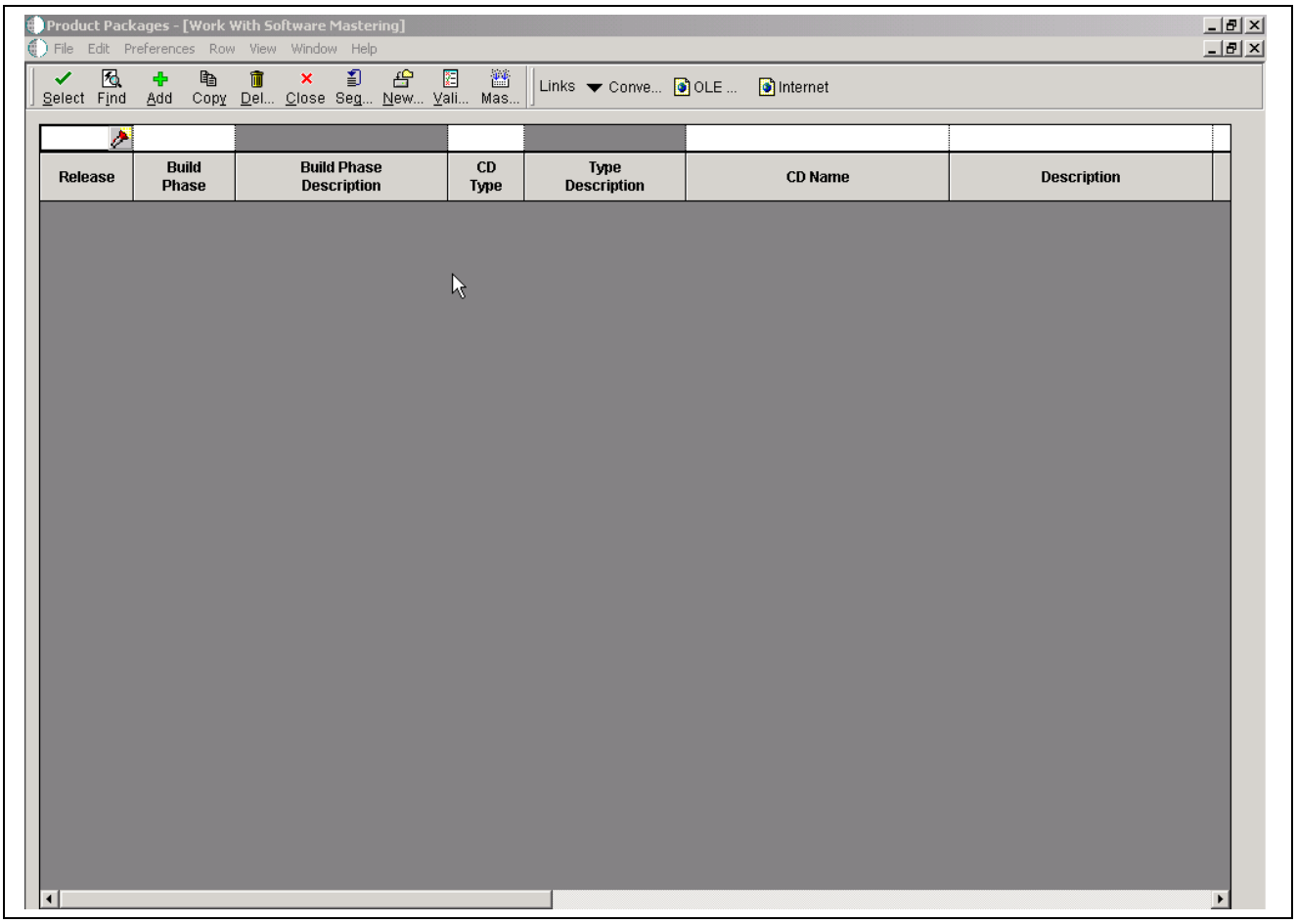
For example:

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

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.



Product Packages [Work with Software Mastering] form

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.

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

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.

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.

Adding a Data Cleanup Procedure

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

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.

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.

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.

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.

Viewing a Table Change Table

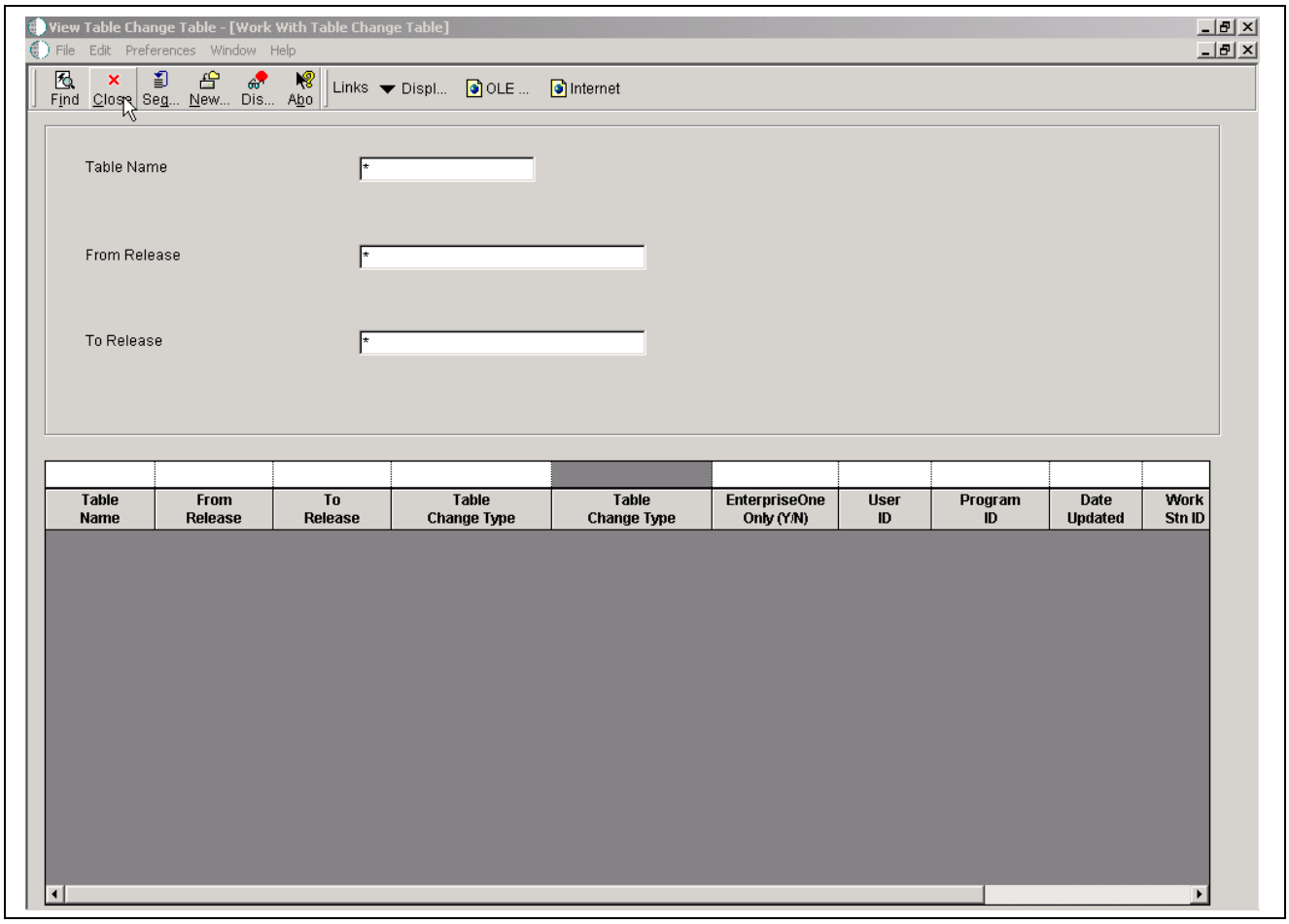
The change table is table that contains all the source table data changes and system metadata necessary to maintain the change table. This function allows you to view the Table Change table.

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.

Viewing the Table Change Table

Access the Work With Table Change Table form.



View Table Change Table [Work with Table Change Table] form

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.

Viewing the Index Change Tables

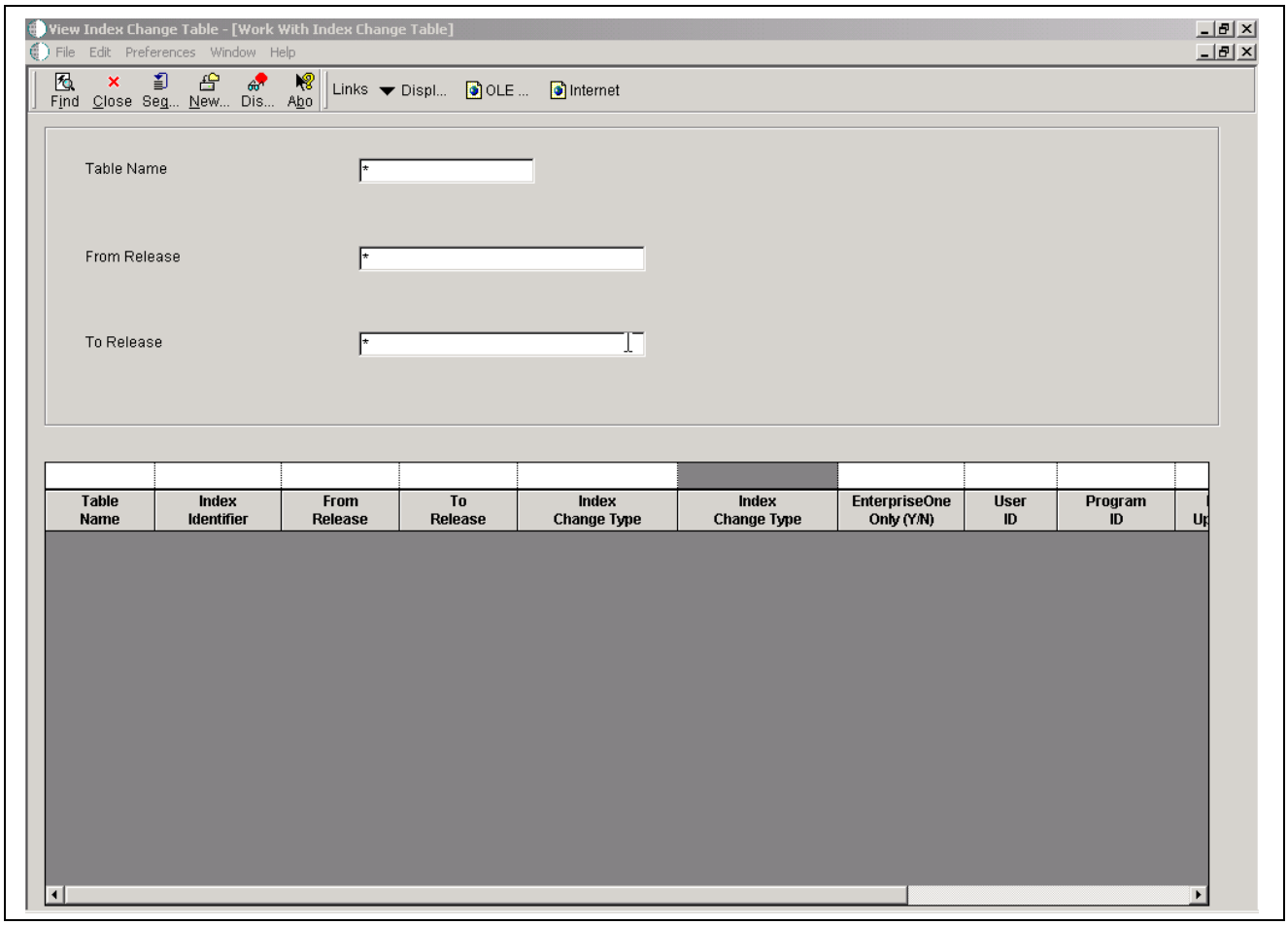
This function allows you to view the index change tables.

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.

Viewing the Index Change Tables

Access the Work With Index Change Table form.



View Index Change Table [Work with Index Change Table] form

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.

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.

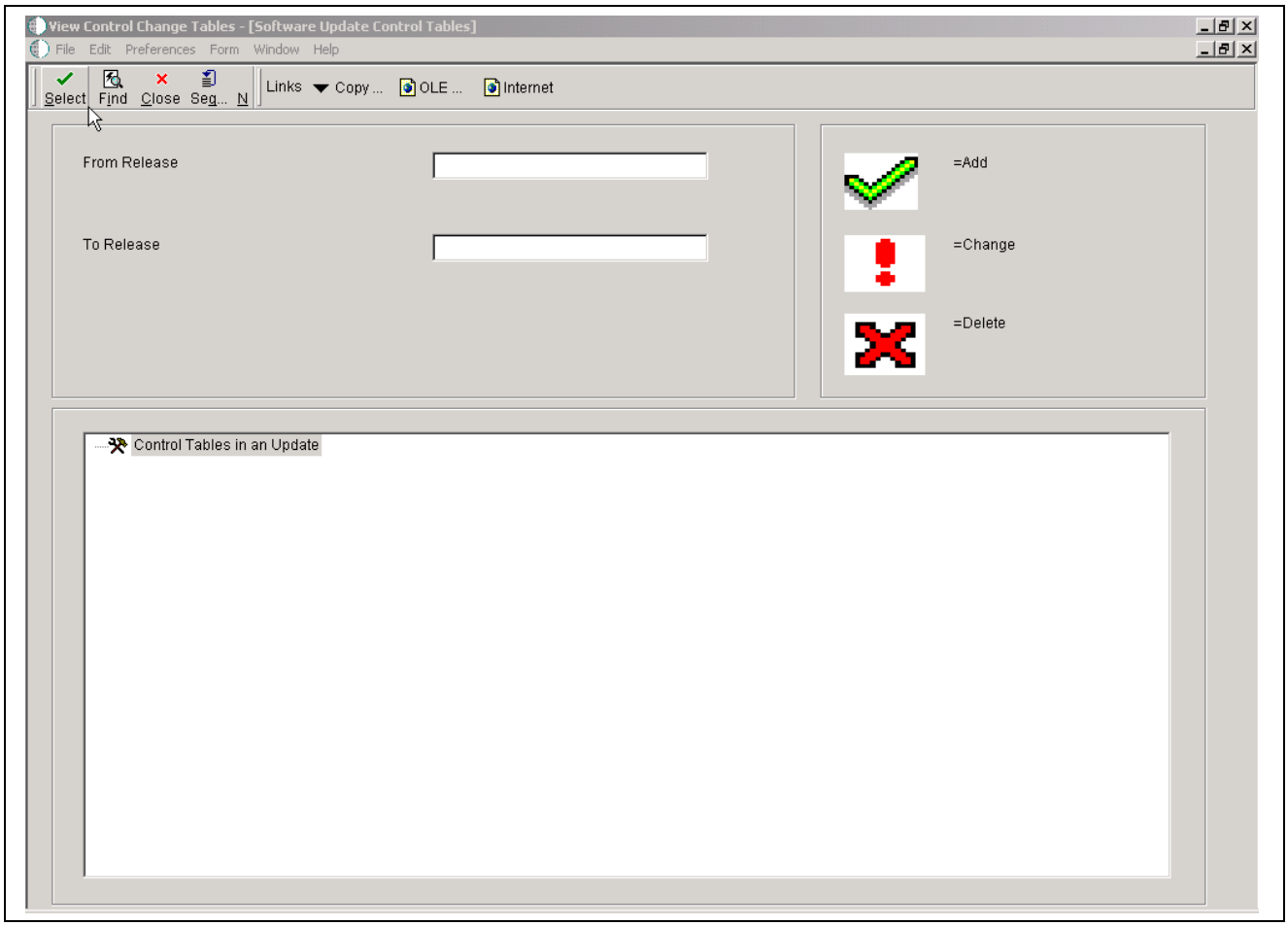
Forms Used with Software Update Control Tables

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.

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.



View Control Change Tables [Software Update Control Tables] form

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.

CHAPTER 7

Working with Translation Updates

This chapter provides an overview of translation updates and discusses how to:

- Create the Translation Update template.
- Collect translation updates.
- Create an update package.

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.

Creating Translation Update Templates

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

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.

Prerequisites

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

Setting Processing Options for the Translation Update Template (R95160)

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

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.

Collecting Translation Updates

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

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.
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.

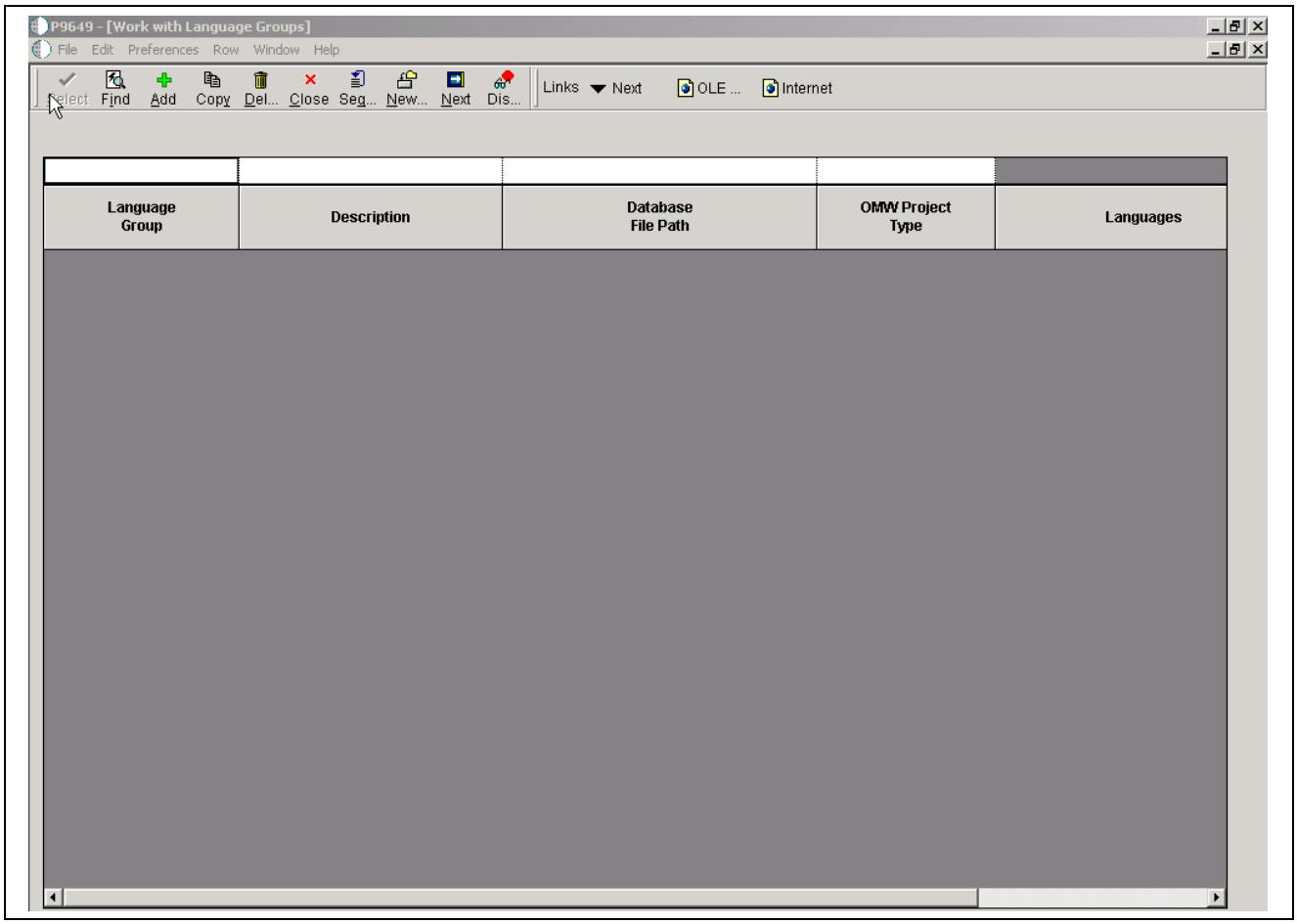
Creating a Translation Updates Database

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.

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.



P9649 [Work with Language Groups] form

Language Group The language group name.

Description A user defined name or remark.

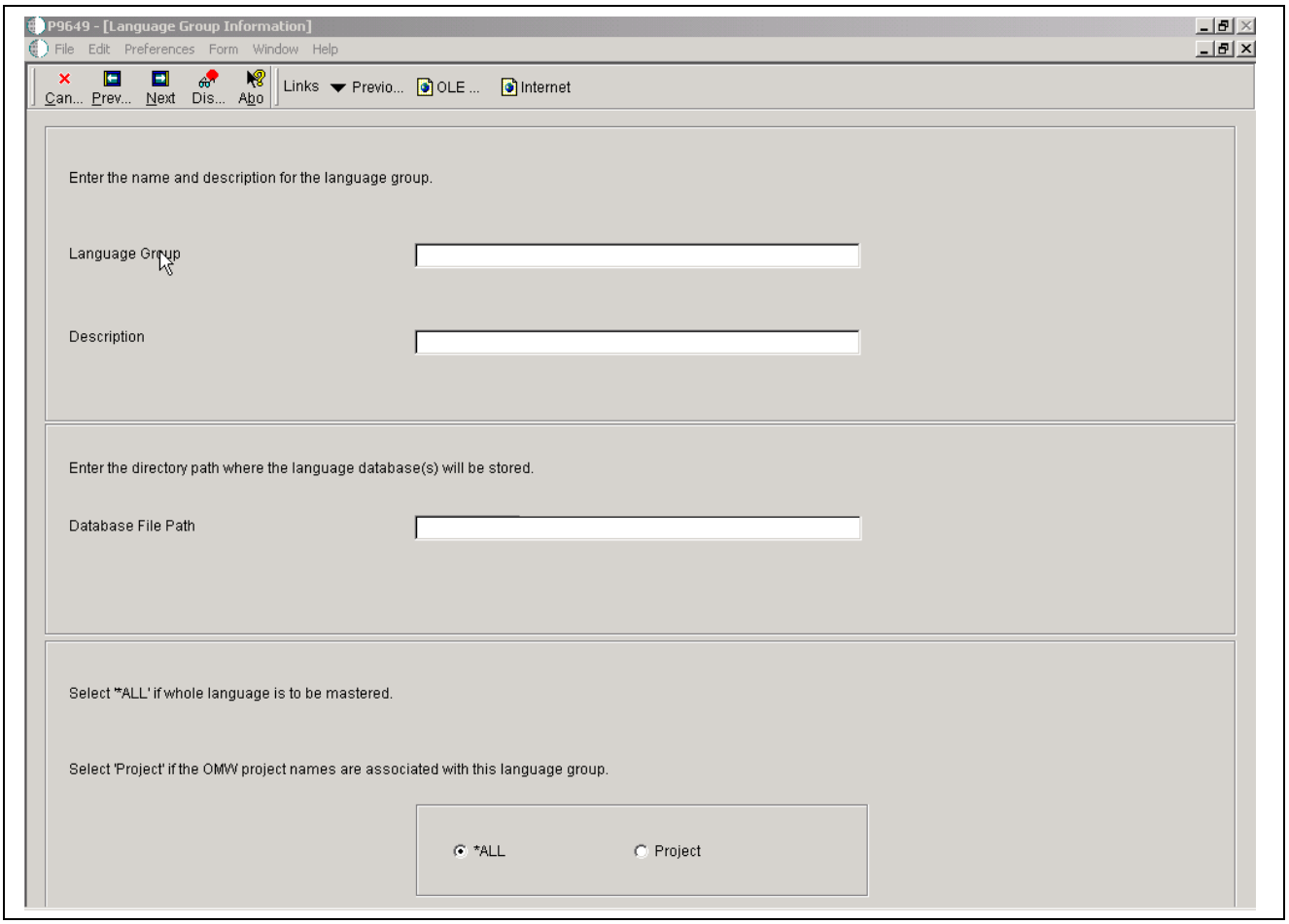
Database File Path An enterprise directory path that you specify, which will be the root directory containing the contents of your software master.

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 top complete a task. All work with objects within enterprise must occur within the context of a project.

Language The language name identifier.

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.



P9649 [Language Group Information] form

Language Group The language group name.

Description A user defined name or remark.

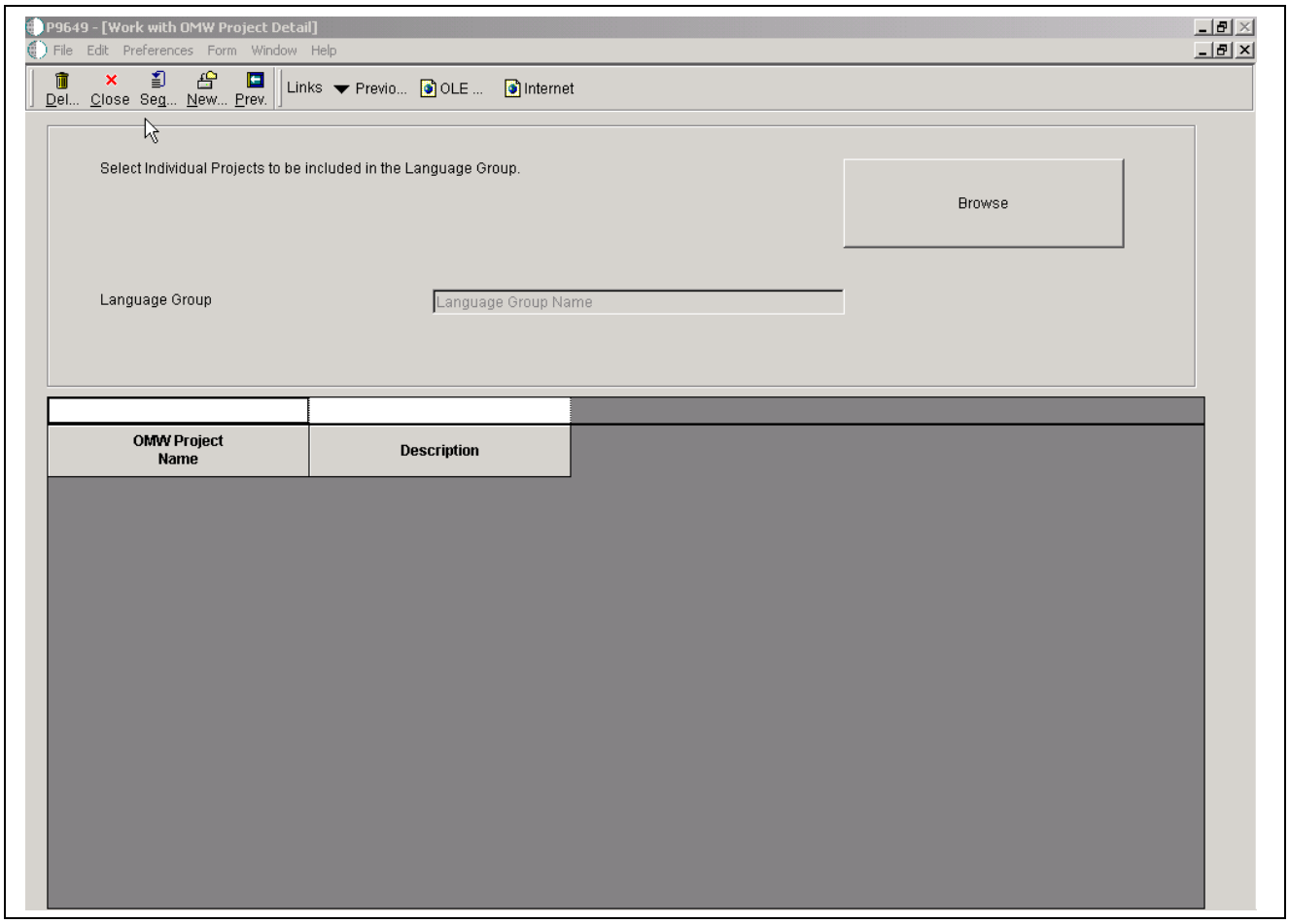
Data File Path An Enterprise directory path that you specify which will be the root directory containing the contents of your software master.

All Select All if whole language is to be mastered.

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

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.



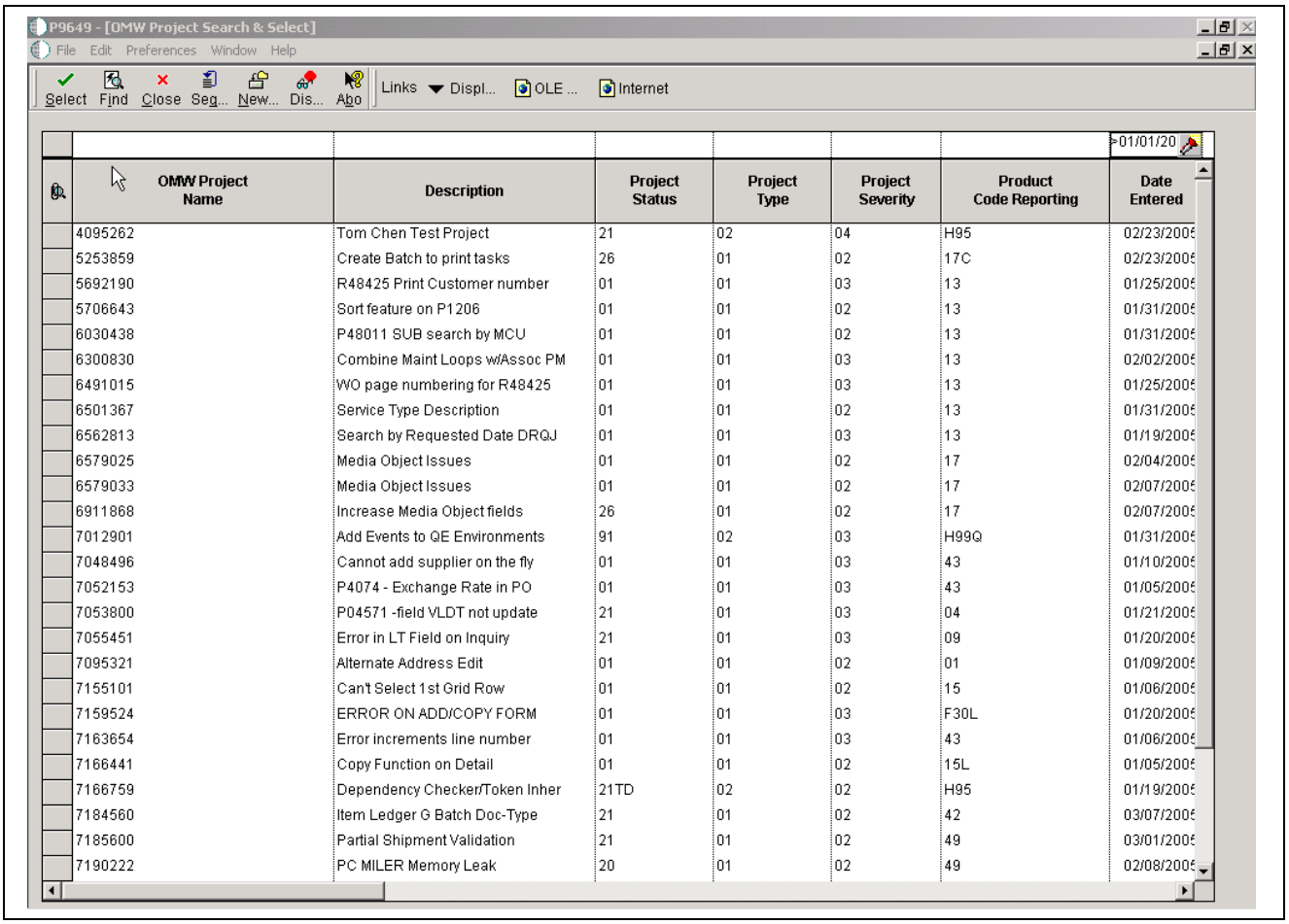
P9649 [Work with OMW Project Detail] form

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.

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.



P9649 [OMW Project Search & Select] form

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

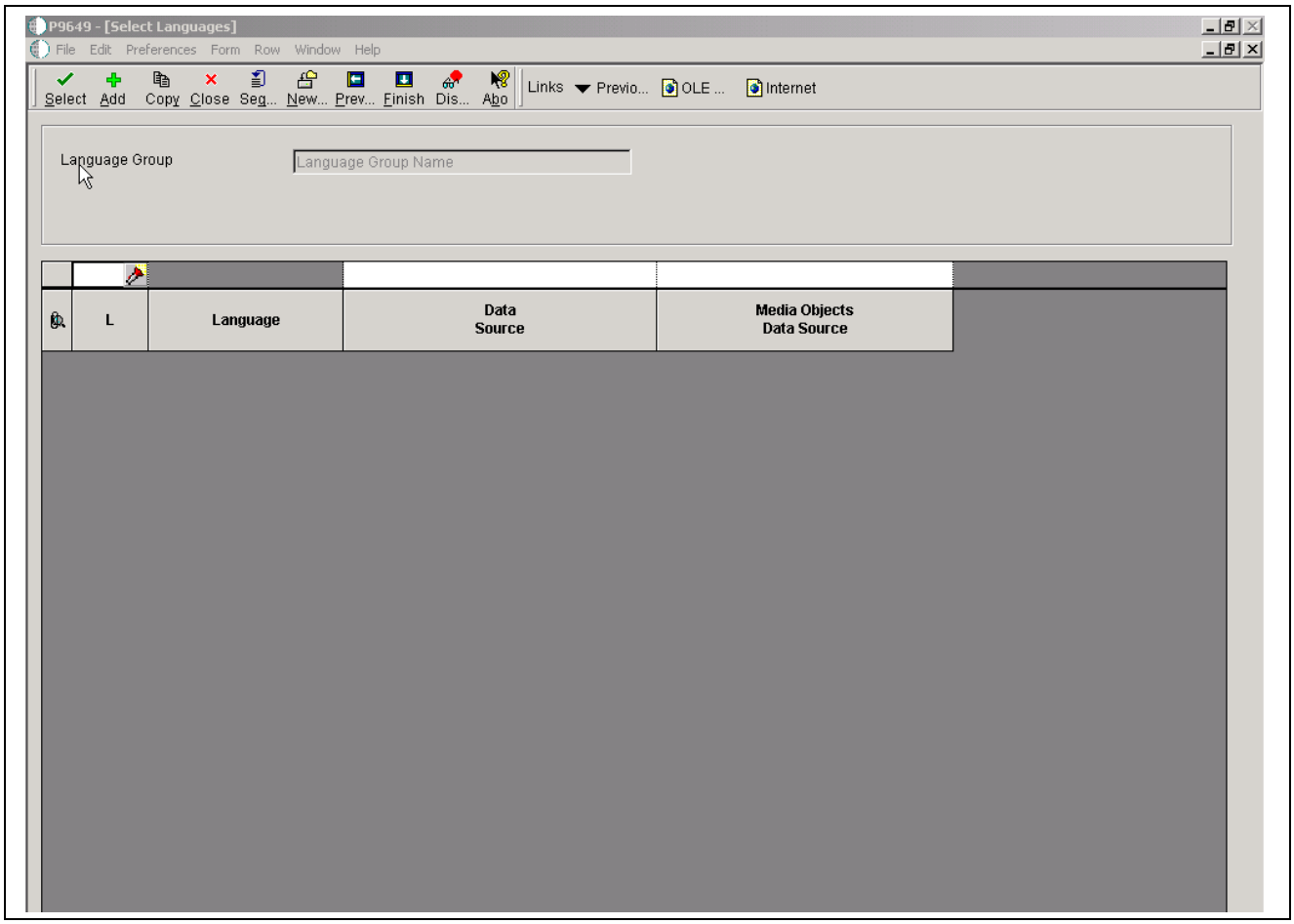
	42 Transfer Prototype to Development *45* Pristine Get *91* Cancelled-Entered in Error This project status should be the starting point in the status activity rules that are defined in the Object Management Configuration Application (P98230).
Project Type	An identifier for a JD Edwards EnterpriseOne 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 top 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.

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.



P9649 [Select Languages] 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.

Creating an Update Package

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

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.

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.

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.

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.

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.

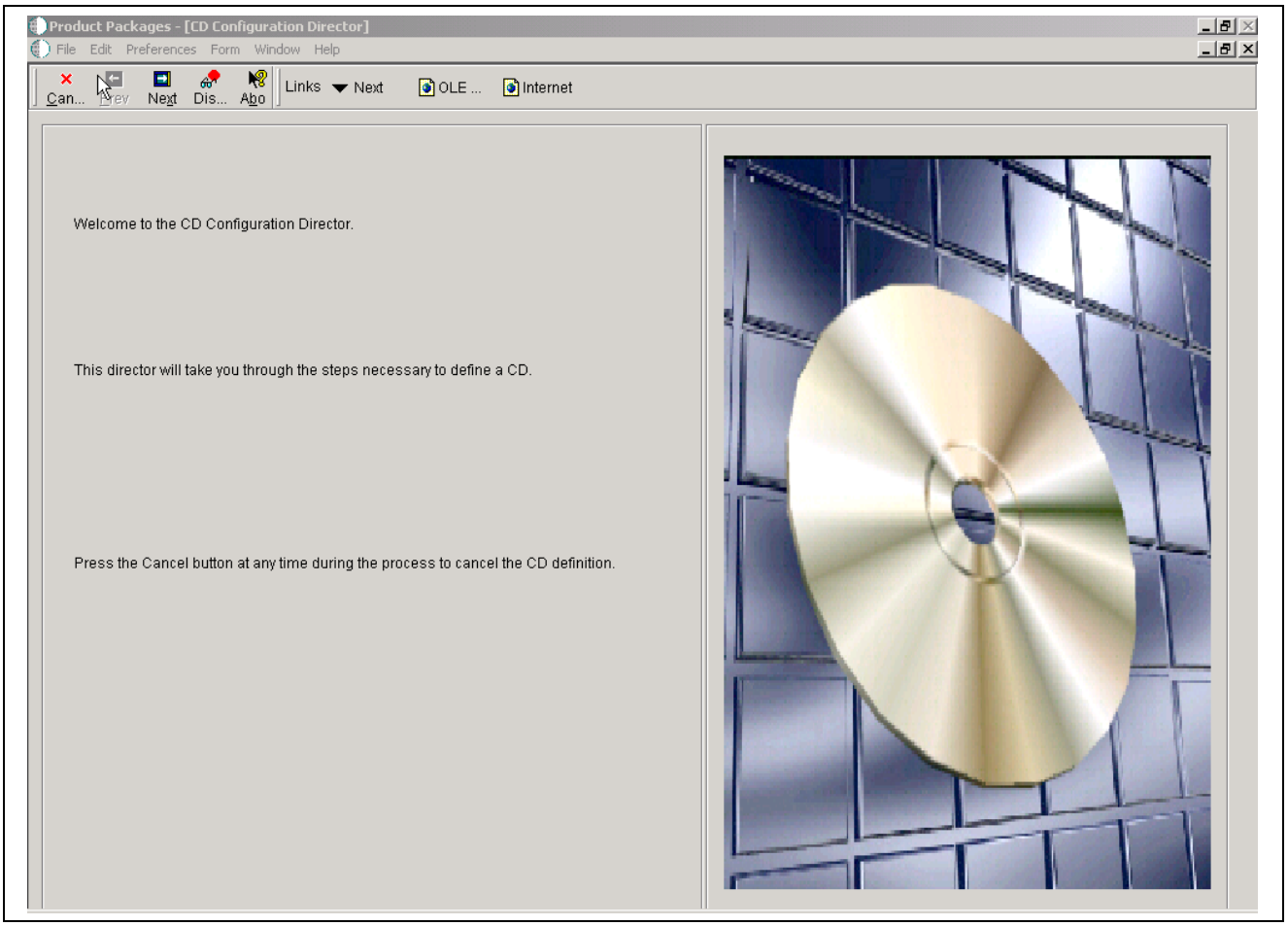
Definitions Date The date the CD was defined.

CD Master Date The date the CD was mastered.

One Off CD Future use.

CD Configuration Director

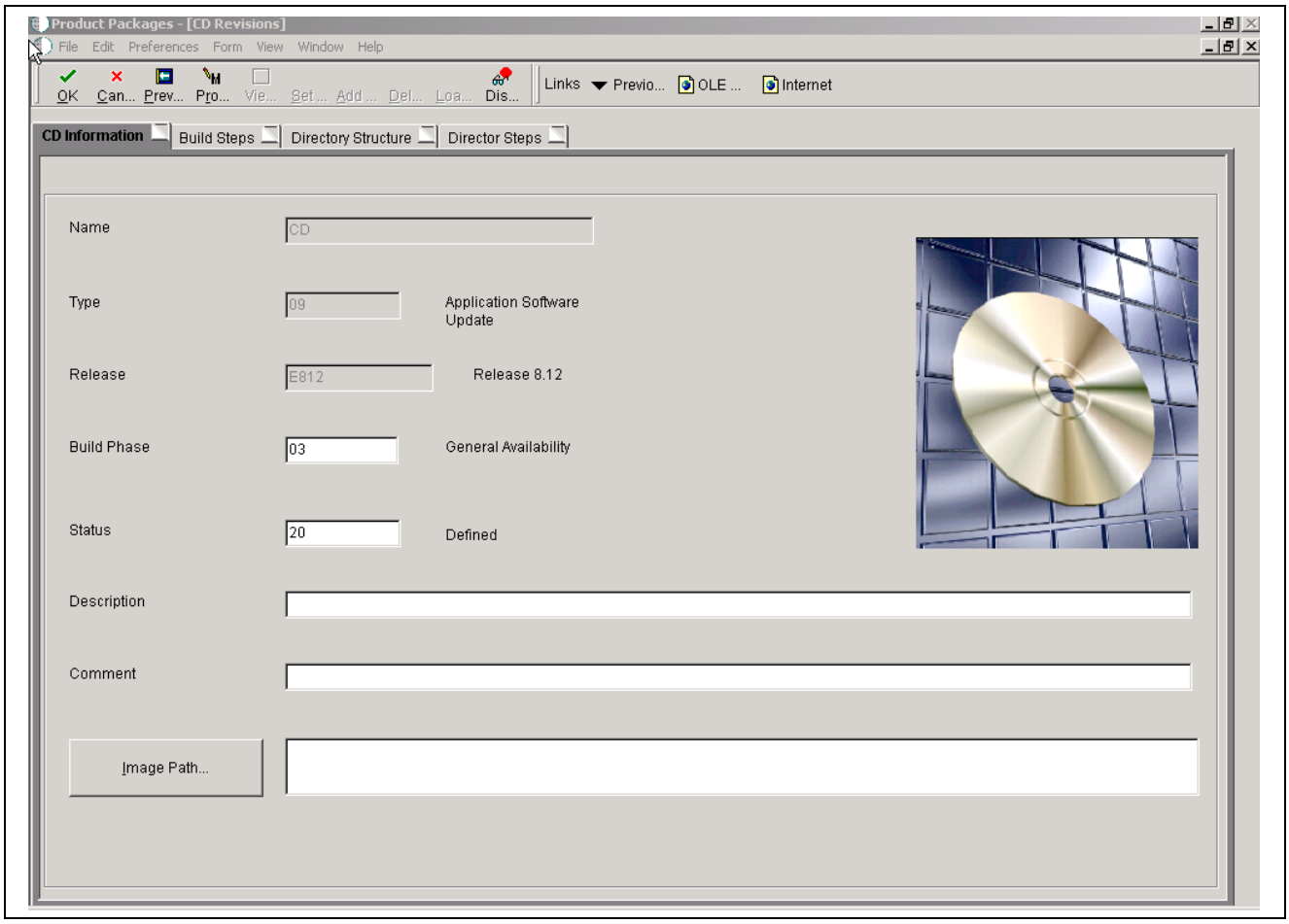
The JD Edwards EnterpriseOne CD Configuration Director guides you through the steps necessary to define a CD.



Product Packages [CD Configuration Director] form

CD Revisions

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



Product Packages [CD Revisions] form

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 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.

APPENDIX A

Troubleshooting Product Packaging

This appendix discusses how to troubleshoot:

- Failure during the check mastering items step.
- Failure to build the feature INF file.
- JDEMasters database not populated.
- Failure to copy change tables.
- Failure to copy custom tables.
- Unable to find product package after installation.
- Error backing up tables in the software update application.
- Problems with Object Librarian records.

Failure During the Check Mastering Items Step

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

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.

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.

Solution

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

JDEMasters Database Not Populated

The build process finishes without error but no tables exist in the JDEMasters database in the $TEMP\master directory. This error occurs when the JD Edwards EnterpriseOne Product Packaging data source is not set up correctly in the database data sources.

Solution

Reconfigure the database data source for JD Edwards EnterpriseOne Product Packaging, and then repeat the build process.

Failure to Copy Change Tables

No change tables (for instance, F960004 and F960005) exist in the JDEMasters 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.

Solution

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

Failure to Copy Custom Tables

Custom tables were not copied to the JDEMasters 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.

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.

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.

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.

Error Backing Up Tables in the Software Update Application

In the JD Edwards EnterpriseOne Software Updates application, after you select the path code and backup, the error message Create Table Failed appears and the application stops processing.

The jde.log file reports this error:

```
JDB9900245 - Failed to find F98611 OWBAK_PRODPACKUD_PY812 in cache
JDB3100011 - Failed to get location of table F983051 for environment JDEPLAN
JDB2100004 - Failed to open table
```

Solution

The JD Edwards EnterpriseOne Software Updates application creates a new data source called OWBAK_*PACKAGE_NAME_PATHCODE*that points to a backup database called Planner\Package*PACKAGE_NAME**PATHCODE*.bak\data\OWBAK_*packagename_pathcode*.mdf and OWBAK_*packagename_pathcode*.ldf. Perform these actions to troubleshoot the problem:

1. Ensure that the latest JD Edwards EnterpriseOne Planner update is installed before you install the product package.
2. Open the Database Data Sources application (P986115) and verify that the data source OWBAK_*PACKAGE_NAME_PATHCODE* is created, where *PACKAGE_NAME* is the product package name in all uppercase letters, and *PATHCODE* is the chosen path code from the Software Updates application. If the data source exists but is not in all uppercase letters, manually delete that data source item and create a new data source item as follows:

Object	Setting
Data Source Name:	*OWBAK_PACKAGE_NAME_PATHCODE*
Data Source Type:	*N - MSDE/ODBC*
Database Name:	*OWBAK_PACKAGE_NAME_PATHCODE*
Server Name:	*(local)*
Platform:	*LOCAL*
Object Owner ID	*dbo*
Database Instance	*JDELocal*
ODBC Data Source Name	*EnterpriseOne Local*

3. Verify that the Open Database Connectivity (ODBC) data source is created and is pointing to a valid location and database.
4. Quit JD Edwards EnterpriseOne, sign in again, and repeat the process.

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.

Solution

After the package is installed on the destination deployment server, open the created JD Edwards EnterpriseOne Product Packaging JDEMasters 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 of JD Edwards EnterpriseOne Terms

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 rational databases, flat file format, and other data or message encoding, such as XML.
alternate currency	A currency that is different from the domestic currency (when dealing with a domestic-only transaction) or the domestic and foreign currency of a transaction. In JD Edwards EnterpriseOne Financial Management, alternate currency processing enables you to enter receipts and payments in a currency other than the one in which they were issued.
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).
as if processing	A process that enables you to view currency amounts as if they were entered in a currency different from the domestic and foreign currency of the transaction.
as of processing	A process that is run as of a specific point in time to summarize transactions up to that date. For example, you can run various JD Edwards EnterpriseOne reports as of a specific date to determine balances and amounts of accounts, units, and so on as of that date.
Auto Commit Transaction	A database connection through which all database operations are immediately written to the database.
back-to-back process	A process in JD Edwards EnterpriseOne Supply Management that contains the same keys that are used in another process.
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 immediate	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. See also direct connect and store-and-forward.
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 categorization A way to categorize business service properties. These properties are categorized by business service.

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).

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.

contra/clearing account A general ledger account in JD Edwards EnterpriseOne Financial Management that is used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations in JD Edwards EnterpriseOne Financial Management.

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.

cost assignment The process in JD Edwards EnterpriseOne Advanced Cost Accounting of tracing or allocating resources to activities or cost objects.

cost component In JD Edwards EnterpriseOne Manufacturing, an element of an item's cost (for example, material, labor, or overhead).

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.

cross segment edit A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced.

currency restatement The process of converting amounts from one currency into another currency, generally for reporting purposes. You can use the currency restatement process, for example, when many currencies must be restated into a single currency for consolidated reporting.

cXML A protocol used to facilitate communication between business documents and procurement applications, and between e-commerce hubs and suppliers.

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.

date pattern A calendar that represents the beginning date for the fiscal year and the ending date for each period in that year in standard and 52-period accounting.

denominated-in currency The company currency in which financial reports are based.

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.

detail information Information that relates to individual lines in JD Edwards EnterpriseOne transactions (for example, voucher pay items and sales order detail lines).

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

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

Do Not Translate (DNT) A type of data source that must exist on the iSeries because of BLOB restrictions.

dual pricing The process of providing prices for goods and services in two currencies.

duplicate published business services authorization records Two published business services authorization records with the same user identification information and published business services identification information.

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 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 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 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.

facility An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. A facility is sometimes referred to as a "business unit."

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.

header information Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows.

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 XBPs. 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.

IServer service This internet server service resides on the web server and is used to speed up delivery of the Java class files from the database to the client.

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.

local standalone BPEL/ESB server A standalone BPEL/ESB server that is not installed within an application server.

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.

matching document A document associated with an original document to complete or change a transaction. For example, in JD Edwards EnterpriseOne Financial Management, a receipt is the matching document of an invoice, and a payment is the matching document of a voucher.

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.

Middle-Tier BPEL/ESB Server A BPEL/ESB server that is installed within an application server.

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

named event rule (NER) Encapsulated, reusable business logic created using event rules, rather that 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.

nota fiscal In Brazil, a legal document that must accompany all commercial transactions for tax purposes and that must contain information required by tax regulations.

nota fiscal factura In Brazil, a nota fiscal with invoice information.

See also *nota fiscal.*

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).

planning family A means of grouping end items whose similarity of design and manufacture facilitates being planned in aggregate.

preference profile The ability to define default values for specified fields for a user-defined hierarchy of items, item groups, customers, and customer groups.

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-grade file server A file server that has been quality assurance tested and commercialized and that is usually provided in conjunction with user support services.

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.

pull replication — One of the JD Edwards EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers using JD Edwards EnterpriseOne data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the F98DRPCN table.

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, such as B73.2 or B73.2.1.

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

Rt-Addressing — Unique data identifying a browser session that initiates the business services call request host/port user session.

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.

quote order — In JD Edwards Procurement and Subcontract Management, a request from a supplier for item and price information from which you can create a purchase order.

In JD Edwards Sales Order Management, item and price information for a customer who has not yet committed to a sales order.

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.

SEI implementation — A Java class that implements the methods that declare in a Service Endpoint Interface (SEI).

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.
Service Endpoint Interface (SEI)	A Java interface that declares the methods that a client can invoke on the service.
SOA	Abbreviation for Service Oriented Architecture.
soft coding	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.
spot rate	An exchange rate entered at the transaction level. This rate overrides the exchange rate that is set up between two currencies.
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.
superclass	An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the superclass of "Oak" and "Elm," for example.
supplemental data	Any type of information that is not maintained in a master file. Supplemental data is usually additional information about employees, applicants, requisitions, and jobs (such as an employee's job skills, degrees, or foreign languages spoken). You can track virtually any type of information that your organization needs. For example, in addition to the data in the standard master tables (the Address Book Master, Customer Master, and Supplier Master tables), you can maintain other kinds of data in separate, generic databases. These generic databases enable a standard approach to entering and maintaining supplemental data across JD Edwards EnterpriseOne systems.
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.

three-tier processing The task of entering, reviewing and approving, and posting batches of transactions in JD Edwards EnterpriseOne.

three-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records to create vouchers.

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.

two-way authentication An authentication mechanism in which both client and server authenticate themselves by providing the SSL certificates to each other.

two-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information.

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.

variance In JD Edwards Capital Asset Management, the difference between revenue generated by a piece of equipment and costs incurred by the equipment.

In JD Edwards EnterpriseOne Project Costing and JD Edwards EnterpriseOne Manufacturing, the difference between two methods of costing the same item (for example, the difference between the frozen standard cost and the current cost is an engineering variance). Frozen standard costs come from the Cost Components table, and the current costs are calculated using the current bill of material, routing, and overhead rates.

versioning a published business service Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

Version 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.

wchar_t An internal type of a wide character. It is used for writing portable programs for international markets.

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 proxy foundation Foundation classes for web service proxy that must be included in a business service server artifact for web service consumption on WAS.

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.

work day calendar In JD Edwards EnterpriseOne Manufacturing, a calendar that is used in planning functions that consecutively lists only working days so that component and work order scheduling can be done based on the actual number of work days available. A work

	day calendar is sometimes referred to as planning calendar, manufacturing calendar, or shop floor calendar.
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

D

F

R

S

T

U

V

W