ORACLE
JD EDWARDS ENTERPRISEONE

JD Edwards EnterpriseOne Tools 8.98 Configurable Network Computing Implementation Guide

September 2008

ORACLE

JD Edwards EnterpriseOne Tools 8.98 Configurable Network Computing Implementation Guide
SKU E1_TOOLS898TCG-B 0908

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

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

Subject to patent protection under one or more of the following U.S. patents: 5,781,908; 5,828,376; 5,950,010; 5,960,204; 5,987,497; 5,995,972; 5,987,497; and 6,223,345. Other patents pending.

Warranty Disclaimer

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

Restricted Rights Notice

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

U.S. GOVERNMENT RIGHTS

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

Hazardous Applications Notice

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

Third Party Content, Products, and Services Disclaimer

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

Open Source Disclosure

Oracle takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in Oracle's JD Edwards EnterpriseOne products and the following disclaimers are provided:

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 downloading 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 Downloading 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 are important to us. We encourage you to tell us what you like, or what you would like to see changed about implementation guides and other Oracle reference and training materials. Please send your suggestions to your product line documentation manager at Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. Or email us at appsdoc@us.oracle.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

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 Configurable Network Computing Implementation Preface

This preface discusses Oracle's JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation.

JD Edwards EnterpriseOne Tools Companion Documentation

Additional information describing the setup and design of the Oracle's JD Edwards EnterpriseOne Tools system resides in companion documentation that apply to many or all 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 8.98, it is highly recommended that you only access and manage these settings for the supported server types using the Server Manager program. See the 8.98 Server Manager Guide on Customer Connection.

The following companion documentation contain information that applies specifically to JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation.

- JD Edwards EnterpriseOne Tools System Administration
- JD Edwards EnterpriseOne Tools Security Administration
- JD Edwards EnterpriseOne Tools Server and Workstation Administration
- JD Edwards EnterpriseOne Tools Package Management
- JD Edwards EnterpriseOne Tools Object Configuration Manager

Customers must conform to the supported platforms for the release as detailed in the JD Edwards EnterpriseOne minimum technical requirements. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the cross-reference material in the Program Documentation at http://oracle.com/contracts/index.html for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

See Also

JD Edwards EnterpriseOne Tools 8.98 System Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools System Administration"

JD Edwards EnterpriseOne Tools 8.98 Security Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools Security Administration"

JD Edwards EnterpriseOne Tools 8.98 Server and Workstation Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools Server and Workstation Administration"

JD Edwards EnterpriseOne Tools 8.98 Package Management Guide, "Getting Started with JD Edwards EnterpriseOne Package Management"

JD Edwards EnterpriseOne Tools 8.98 Object Management Workbench Guide, "Getting Started with JD Edwards EnterpriseOne OMW"

CHAPTER 1

Getting Started with JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation

This chapter discusses:

- Configurable Network Computing overview
- Configurable Network Computing implementation

Configurable Network Computing Overview

Oracle's JD Edwards Configurable Network Computing is an application architecture that enables interactive and batch applications, composed of a single code base, to run across a network of multiple server platforms and databases. The applications consist of reusable business functions and associated data that can be configured across the network dynamically. The overall objective for businesses to provide a future-proof environment that enables them to change organizational structures, business processes and technologies independently of each other.

Configurable Network Computing Implementation

Oracle's 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 pre-integrated and share a common database, which reduces the implementation process, minimizes ongoing administration, and provides customers the flexibility to add in new applications, modules, and tools as needed.

A complete list of these resources appears in the preface in *About This Documentation* with information about where to find the most current version of each.

CHAPTER 2

Understanding Configurable Network Computing Implementation Teams

This chapter provides an overview of implementation teams and discusses:

- Technology roles
- Development roles
- Functional roles
- System Integration roles

Implementation Teams

The Oracle JD Edwards EnterpriseOne implementation methodology defines specific roles that are involved in the design, installation, and configuration of an ERP solution. These roles are generally divided into four implementation teams:

- Technology - installation and upgrades, system administration, security, change management
- Development - data conversions, interfaces, custom modifications
- Functional - business process, application configuration, integration and testing, end-user training
- Systems Integration - data center design, hardware support, network infrastructure, third-party software

Each of these implementation teams is staffed by both consultant and customer roles. As an implementation progresses to completion, the consultant roles diminish, while the customer roles remain and often increase in level of responsibility. It is critical, therefore, that the customer ensures that each role to be assumed by its personnel is adequately trained.

Technology Roles

Typically, the technology project team is led by a single consulting role, the technology specialist, and two customer roles, the system administrator and the change management administrator. The technology specialist and system administrator are involved with installing JD Edwards EnterpriseOne and setting up environments, users, security, distributed processing, data replication, and other system administration and operations support topics. The technology specialist and change management administrator are responsible for setting up version control, applying software updates and service packs, reviewing and promoting code and data across change management environments, and deploying code and data changes to the servers and workstations involved in the ERP solution.

Development Roles

The development project team is typically led by a custom solution consultant and staffed by one or more application developers. The custom solution consultant resolves business issues by developing applications. Primary responsibilities include developing a data migration strategy, designing interfaces to legacy and third-party systems, and designing custom modifications with upgrades in mind. The application developers migrate legacy data, code and test interfaces, code and test custom modifications, and integrate all code changes into the ERP solution.

While the change management administrator performs the version control functions that control the acceptance, promotion, and deployment of software changes, the custom solution consultant must help develop the internal procedures for realizing an application development life cycle within your business. In addition, development team members must be aware of change management tools and procedures, as well as how the technology components affect the design and operation of interfaces and custom modifications.

Functional Roles

The functional project team is led by a consulting project manager and a customer project manager, and staffed by application specialists and customer process owners. These project members are responsible for the design, configuration, and deployment of JD Edwards EnterpriseOne applications, as well as the modeling of all business processes that will be realized through the application set. After JD Edwards EnterpriseOne is installed, configured, and rolled out, the application specialists continue in their role as product experts. Although application specialists do not implement technology-level solutions, they must understand how the software handles distributed processing, data replication, environments, and so on, because these application issues influence technology design and configuration. In addition, application specialists and process owners must become expert at troubleshooting potential problems and identifying the difference between a technology issue and an application issue.

Systems Integration Roles

The systems integration project team is responsible for many tasks that are outside the scope of services. Third-party consultants provide some of these services and supplement JD Edwards EnterpriseOne staff as CNC consultants, network architects, custom modification consultants, and so on. In addition, customers provide hardware and network infrastructure support.

Implementing the JD Edwards EnterpriseOne system includes many tasks that are outside the scope of JD Edwards EnterpriseOne software and services. Systems integration (that is, third-party) consultants provide these services to help you align the infrastructure to optimally support JD Edwards EnterpriseOne applications and runtime services, as well as expand the overall business solution with complementary third-party products. These consultants are able to assist with such services as data center design, IT process improvement, and network infrastructure. They are also able to assist with the installation, configuration, and integration of third-party hardware and software products that enhance and extend the JD Edwards EnterpriseOne software solution. These project members should be aware of the architecture and technical behavior of JD Edwards EnterpriseOne software and of how the various technology components interact with operating systems, database management systems, third-party middleware, and the network.

CHAPTER 3

Understanding Configurable Network Computing Foundation

This chapter provides an overview of configurable network computing and discusses:

- Configurable Network Computing advantages
- Configurable Network Computing fundamentals

Configurable Network Computing Foundation Overview

Oracle's JD Edwards Configurable Network Computing is the technical architecture for Oracle's JD Edwards EnterpriseOne software. Configurable Network Computing enables highly configurable, distributed applications to run on a variety of platforms without users or analysts needing to know which platforms or which databases are involved in any given task. Configurable Network Computing insulates the business solution from the underlying technology. Enterprises can grow and adopt new technologies without rewriting applications.

JD Edwards EnterpriseOne software comprises these software components:

Design Tools Design Tools provides a unified set of tools to create all interactive applications, batch applications, and reports.

Applications Applications provides the interactive and batch applications that perform your business needs. For example, Purchase Order Entry and General Ledger Post are applications.

Software Foundation Code Software Foundation Code provides underlying core processing that both interactive and batch applications depend on in order to run.

Software Middleware Software Middleware provides middleware that insulates the applications from the underlying database, operating system, hardware, messaging systems, and telecommunications protocols. Middleware insulates your business solution from the platform technology.

Configurable Network Computing Advantages

This section discusses the advantages that the JD Edwards EnterpriseOne Configurable Network Computing architecture provides:

- Network-centric software

- Flexible and leveraged technology
- Worldwide business support
- Custom solutions without consequences

Network-Centric Software

Network-centric software enables you to create a uniform interface that supports a multiple-platform network. This compatibility across platforms provides:

Immediate availability of enhancements to all supported applications. Changes to these items are reflected in applications across the network:

- Business objects
- Business rules
- Modes of processing
- Hardware and database
- Browser interface to support for internet technology

JD Edwards EnterpriseOne platform-neutral business specifications, or middleware, that comprise a common set of Application Program Interfaces (APIs) that integrate multiple-vendor, multiple-protocol differences. This integration insulates developers from the need to program to a specific platform.

Flexible and Leveraged Technology

You create the applications using tools that do not require a designer to master a programming language. JD Edwards EnterpriseOne tools conceal the code and enable the designer to concentrate on creating applications that are specific to current business needs and accommodate changes to business rules without reprogramming the application source code.

JD Edwards EnterpriseOne is object-based and event-driven to provide you with more efficient business processes. Developers can reuse objects between applications for different purposes. This reusability provides consistency throughout all JD Edwards EnterpriseOne applications.

JD Edwards EnterpriseOne does not rely on one command or keystroke to process information; rather, it processes information at strategic moments during the use of an application. For example, when a user moves among fields on a form, the system processes the information at the moment when the cursor leaves the field. JD Edwards EnterpriseOne immediately notes any errors and hides processing, such as an update of files that might also store information for the field, when the user moves to the next field on a form.

In addition, JD Edwards EnterpriseOne provides a common interface between applications. When you move from form to form, you see the same general setup.

Worldwide Business Support

JD Edwards EnterpriseOne provides support for mixed currency and languages. Also, you can run JD Edwards EnterpriseOne on platforms from servers to laptops. This scalability enables a traveling consultant to interface with the system and enter records. The consultant can then send these updated records over the internet to keep files as current as possible.

Note. As of the ERP 8.0 release of JD Edwards EnterpriseOne software, JD Edwards EnterpriseOne no longer coexists with WorldSoftware. Contact Oracle for more information about migrating from WorldSoftware A73 to JD Edwards EnterpriseOne.

Custom Solutions Without Consequences

You can make custom solutions to business applications with few or no consequences when you upgrade to a new release of JD Edwards EnterpriseOne. The JD Edwards EnterpriseOne toolset acts as an idea enabler by enabling you to transform a concept into a viable business solution. You maintain consistency across the enterprise, retain flexibility to adapt to changing business requirements, and minimize the time required to implement upgrades. This list provides examples of areas in JD Edwards EnterpriseOne that you can customize without consequences during an upgrade:

- Vocabulary overrides
- User overrides
- Versions
- Processing options
- Code generator options

Configurable Network Computing Fundamentals

The section discusses the fundamentals of the Configurable Network Computing architecture, which consists of these items:

- Environments
- Path codes
- Data Sources
- Object Configuration Manager (OCM)
- Object storage
- Object deployment

Environments

An JD Edwards EnterpriseOne environment is a collection of pointers indicating the location of data and JD Edwards EnterpriseOne software objects. An environment answers these questions:

- Where is my data?
- What machine will process my logic?
- What directory contains the object being processed?

JD Edwards EnterpriseOne provides an environment as a pointer to data and logic objects. For example, in the Purchase Order application the answers are as follows:

Question	Response
Where is my data?	A user clicks the Find button to locate a Purchase Order. The environment determines in which database the table resides.
What machine will process my logic?	When finished entering an order, the user clicks OK. The environment determines where the logic (a master business function) necessary to record the transaction will process and where the transaction tables reside to enter the order.
What directory contains the object being processed?	After entering a user ID and password, a user must select the environment to log on to. If you have multiple sets of objects, selecting the environment determines which objects that JD Edwards EnterpriseOne executes (the directory in which they reside). This location is called a path code, and JD Edwards EnterpriseOne defines it in the Library List Master File (F0094) table.

Path Codes

A path code can refer to the central development objects on the deployment server or to replicated objects on a workstation or logic server. A path code exists for each unique set of central objects. For example, you might have a set of objects reserved for software updates that you can deploy to users and a set of objects that you reserve for major enhancements.

A set of objects or the path code can reside in these locations:

Central Server Contains the central set of development objects specifications. All development occurs in this location. The path code connects the specifications and the C components on the deployment server.

Workstation Contains a replicated set of objects that JD Edwards EnterpriseOne uses at run time.

Shared Object Server Contains a replicated set of objects that JD Edwards EnterpriseOne Enterprise and JAS servers use to process logic on these servers.

The Object Path table (F00942) contains path codes that track a set of objects and their location within JD Edwards EnterpriseOne.

Data Sources

A data source is the specific location of data or distributed processing. JD Edwards EnterpriseOne data sources can be:

- An entire database in a specific location, regardless of the type of database, such as a MSDE located in a specific directory or a library in DB2/400
- A specific machine in the enterprise that processes logic

The platform and data sources work together. You must define both the server that processes the logic and the databases that store the data. If multiple databases within one database management system (DBMS) reside on a machine, you must define each database to JD Edwards EnterpriseOne.

Do not confuse Microsoft open database connectivity (ODBC) data sources with JD Edwards EnterpriseOne data sources. The ODBC data source defines databases to various third-party communication products such as Client Access, Rumba, SQL Server, and MSDE. JD Edwards EnterpriseOne data sources define both databases and logic servers to JD Edwards EnterpriseOne.

This list describes JD Edwards EnterpriseOne data sources that you might use in the configuration:

Oracle DBMS A JD Edwards EnterpriseOne data source for an Oracle DBMS points to an Oracle Connect String and a Table Owner.

Oracle OEE A JD Edwards EnterpriseOne data source for an Oracle DBMS points to an Oracle Connect String and a Table Owner.

SQL Server DBMS A JD Edwards EnterpriseOne data source for a SQL Server DBMS points to a SQL Server Database (ODBC data source) and a Table Owner.

DB2/OS400 DBMS A JD Edwards EnterpriseOne data source for a DB2/OS400 DBMS points to a RDB directory entry and a Library (ODBC data source).

MSDE DBMS A JD Edwards EnterpriseOne data source for a Microsoft Data Engine (MSDE) DBMS points to a MSDE database (OLBC data source).

Object Configuration Manager

The Object Configuration Manager (OCM) program (P986110) is a tool that configures distributed processing and distributed data at runtime without requiring programming. Using the Object Map table, the OCM points to the correct data, batch process, or business function for a given environment and user. The OCM is the control center for the runtime architecture. JD Edwards EnterpriseOne always uses the OCM to locate the data and platform needed to execute the distributed logic.

Every environment has an associated set of OCM mappings that indicate the distributed data and distributed processing locations for that environment.

This equation represents the relationship among the OCM, a path code, and an environment:

ENVIRONMENT = PATH CODE + OCM MAPPINGS

Where:

Path Code = what directory contains the object being processed

OCM mappings = (what database stores the data) + (where should the logic object execute)

Object Storage

JD Edwards EnterpriseOne provides three general storage formats; central objects, package objects, and serialized objects to accommodate several functions in JD Edwards EnterpriseOne.

Central Objects

You store objects in a central location to enable for these:

- Deployment
- Redeployment
- Development

Central objects consist of object specifications for each JD Edwards EnterpriseOne object and C components for code-generated objects. Store the central object specifications in a relational database on either a deployment server or an enterprise server, depending on available resources. Store C components for code-generated objects in directories on the deployment server.

To deploy objects out to the enterprise, you define a package that JD Edwards EnterpriseOne creates from central objects. Each package contains a copy of the central objects. This copy consists of object specifications, and linked and compiled C components.

Package Objects

A package contains the necessary specifications and function libraries to run the business applications. Win32 clients have their own dedicated packages while servers share a single package. For example, to execute the Address Book application on a workstation, the workstation needs the object specifications and the compiled dynamic link library for the Address Book application and for any object that the application uses, such as data dictionary items, tables, and business views. The workstation and Enterprise server will store the compiled libraries on its file system. The object specifications will be stored in a Spec package.

An JD Edwards EnterpriseOne server shares a spec package contained in an enterprise RDMS with other JD Edwards EnterpriseOne servers and other web servers. An JD Edwards EnterpriseOne workstation now has a local database instead of TAM files.

Serialized Objects

The web server uses on-demand generation to create serialized objects from the shared object package when needed at runtime. The generator turns JD Edwards EnterpriseOne specifications into Java code, which enables you to access JD Edwards EnterpriseOne applications in HTML. The JD Edwards EnterpriseOne forms and applications that are generated are HTML objects. JD Edwards EnterpriseOne stores the objects in the local database and retrieves them at runtime. The serialized objects serve the function of a persistent cache.

Object Deployment

Deploy JD Edwards EnterpriseOne to the workstations and servers using any of these methods:

- Initial installation, for workstations and servers.
- Workstation installation, for workstations.
- Application installation, for workstations.
- Just-in-time installation, for workstations.

Initial Installation

The installation process is based on a centralized deployment server model. The Deployment Server Installation program (P986115) copies JD Edwards EnterpriseOne installation software from the CD-ROM to the deployment server. From the deployment server, you redistribute the software to the enterprise servers and workstations.

Workstation Installation

The Workstation Installation program (P986115) retrieves software from the package that you request. A package contains instructions that describe where to find the necessary components that the Workstation Installation program deploys to the local computer.

Each package represents a record of the central objects at a point in time. Once you build and test a package, you can safely modify central objects because users will not receive those objects until you build another package and make it available to them. Building a package involves copying the central objects to the package itself. The package then contains replicated objects, which JD Edwards EnterpriseOne can read at runtime.

Application Installation

Application installation can be used to quickly deploy changes to an individual application. The workstation initiates the application installation, and the deployment server responds by gathering and delivering all objects that are necessary to run the application.

Advantages of application installation are:

- You do not need to build a new package and perform a global build before deploying the application change.
- Developers and testers can use application installation to load changes that were recently checked into the central objects onto their machine.

Just-in-Time Installation

Just-in-time installation installs applications to the workstation the first time you use them. For example, when you deploy a custom menu that contains a new application to a workstation, the object automatically installs on the workstation when a user clicks the menu option for the application.

Server Deployment

Server deployment has been modified due to the migration from TAM specs to XML. A major change to server deployment are two new deployment models available for Java called the Discovery Process and the Spec.ini override. The Discovery Process is a web server auto-discovery model which places the system in control of the deployment.

Enterprise Server deployment

JD Edwards EnterpriseOne is a multi-tier system that executes “Applications”. The applications logic is contained in a “Package”. These packages are built and deployed on “Nodes”. Nodes are the participants in the system; such as a Windows client, Enterprise server, Java node (for example: JAS, RTE server), and so forth.

The Spec.ini is a new file that is deployed to the \spec directory when a full package is installed. This file points to an XML package in a database.

The different deployments by releases are:

- Deployment prior to 896:
  - Specs are in TAM binary format.
  - Specs are stored on the local file system.
  - A tool (“eGenerator”) is used to convert TAM specs to serialized objects.
  - Generation is manual and needs to be done every time a package is deployed.
  - Generation only from a Windows client
- Deployment in 896:
  - TAM Deployment (8.10, 8.11) is still supported. It uses the same deployment model as prior service packs.
  - H4A special cases.
  - XML Deployment (starting with 8.12) for Windows client, Enterprise server, Java nodes.

- Deployment in 896 (H4A).
  - No change. Specs will be generated from the local package, in TAM or XML. No configuration changes required for Metadata.
  - Configuration flags in jdbj.ini will be ignored. specGenerateOnDemand is ignored and considered true.
- Deployment in 896 (XML)
  - For all nodes the Specs are stored in XML in a RDBMS.
  - The Windows Client uses the local MSDE database with XML specs. The Spec.ini file is located in the \spec folder and points to the local database. It is deployed when a full package is installed on the fat client.
  - The Enterprise Server Spec.ini is deployed to the \spec directory when a full package is installed and points to an XML package in a database.
- Java node (JAS Server, RTE server, and so forth).

Java Node auto-discovery

JD Edwards EnterpriseOne Java nodes utilizes a new deployment model called Discovery Process which enables the system to be in charge of controlling the deployment. Deployment of a package is fully automated. This process increases integrity and is best suited for production environments.

The web server Discovery Process will:

- Locate the “default” enterprise server. The “default” server is defined as the default BSFN server for the signed-on user.
- Find what package is deployed on that server.
- Find the content of the package (including incremental package updates).
- Delete any obsolete serialized objects.
- Generate serialized objects on demand.

Some of the benefits to the Discovery Process deployment are:

- Full and update packages are detected and applied automatically.
- Serialized objects are cleared when invalid.
- The web server executes application logic which is always up to date with the deployed package.
- No manual process involved.
- No need to bounce servers.
- No need to deploy explicitly to a web server node.

CHAPTER 4

Understanding Path Codes

This chapter provides an overview of understanding path codes and path code usage.

Understanding Path Codes

A path code is a pointer to a set of objects. For each set of objects in the configuration, you must define a path code in the Object Path Master File table (F00942).

Understanding Path Code Usage

Path codes are used for installation, runtime, and development for each set of objects in the configuration, you must define a path code in the Object Path Master File table (F00942).

Path Codes at Installation

You must define a path code in the Object Path Master File table (F00942) for each set of central objects. A set of Oracle's JD Edwards EnterpriseOne objects consists of a central-objects data source and a directory of objects, which includes business function source and include files, object files, and dynamic link libraries (DLLs). A path code definition contains the data source name of the central-object specifications and the directory path to the objects.

When you build a package for the workstation, you must specify a path code. The software uses this path code to determine which set of central objects to use as the source for the package and the directory to use as the destination for the package.

If the software opens an application that does not reside on the workstation and Just-In-Time Installation (JITI) is set for the workstation, the deployment data source will install the needed objects to the workstation at runtime.

Path Codes at Runtime

JD Edwards EnterpriseOne uses path codes at runtime in these ways:

- To validate available environments.

  When you log on to JD Edwards EnterpriseOne, the system checks the path codes that you have defined in your environments against the path code directories that are physically installed on the workstation. If that workstation does not have a path code that you defined in one of your environments, that environment is not displayed when you log on.

- To determine the directory location of a requested object.

Path Codes at Development

When you check out an object for development, you use the Oracle's JD Edwards Object Management Workbench to specify a path code. The software uses the path code to determine where the central objects are stored and checks out the object from those locations (both database and file server).

When you check in an object, you use the Object Management Workbench to specify a path code. The software uses the path code to determine the location of the central objects in which to place the objects.

Understanding Path Code Definitions

Path codes keep track of sets of objects and their locations in JD Edwards EnterpriseOne. For every set of objects in the configuration, JD Edwards EnterpriseOne requires a path code definition in the Object Path Master File table (F00942).

JD Edwards EnterpriseOne recommends a separate path code definition for each of the these sets of objects:

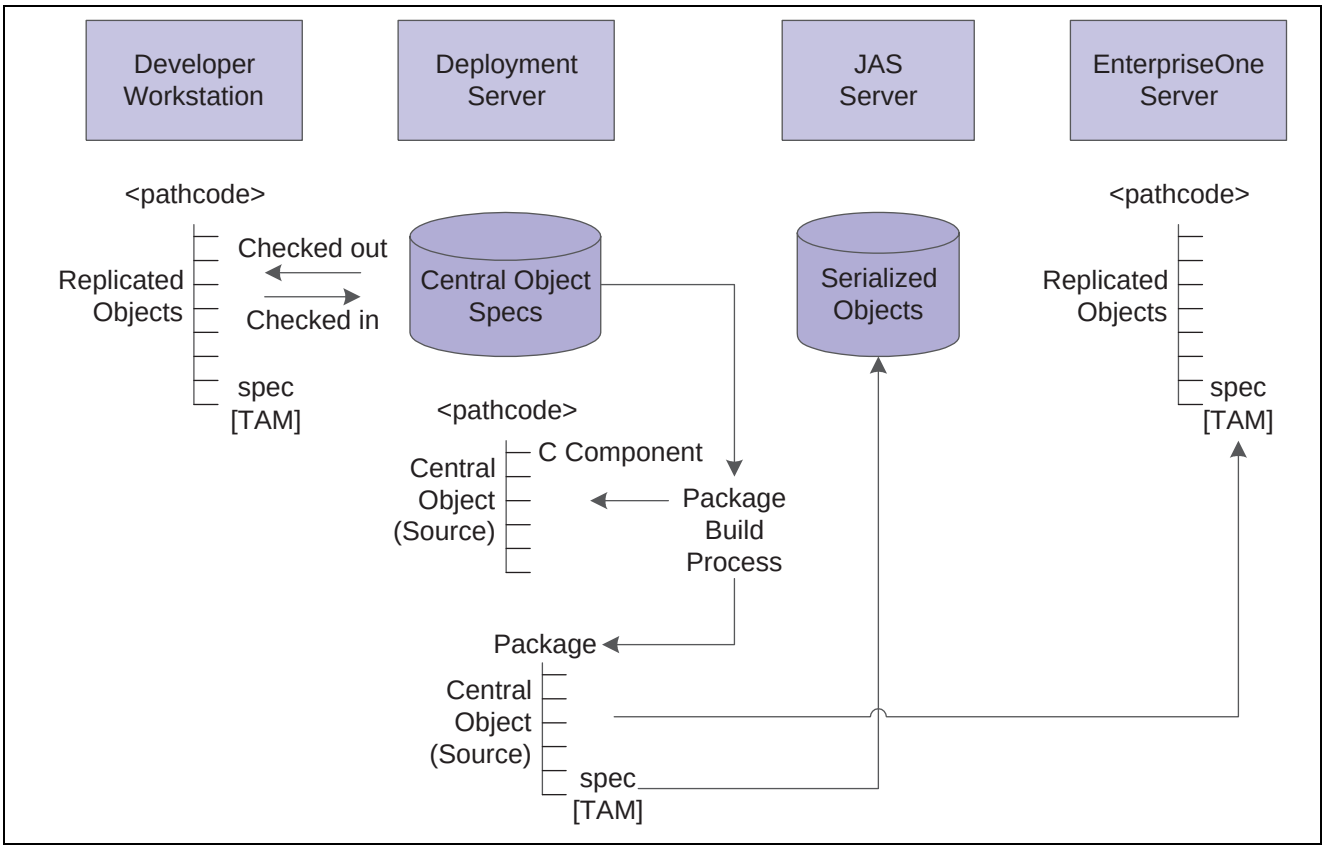
- Pristine objects.
- Production objects.
- Development objects.
- Prototype objects.

Path Codes and Object Storage

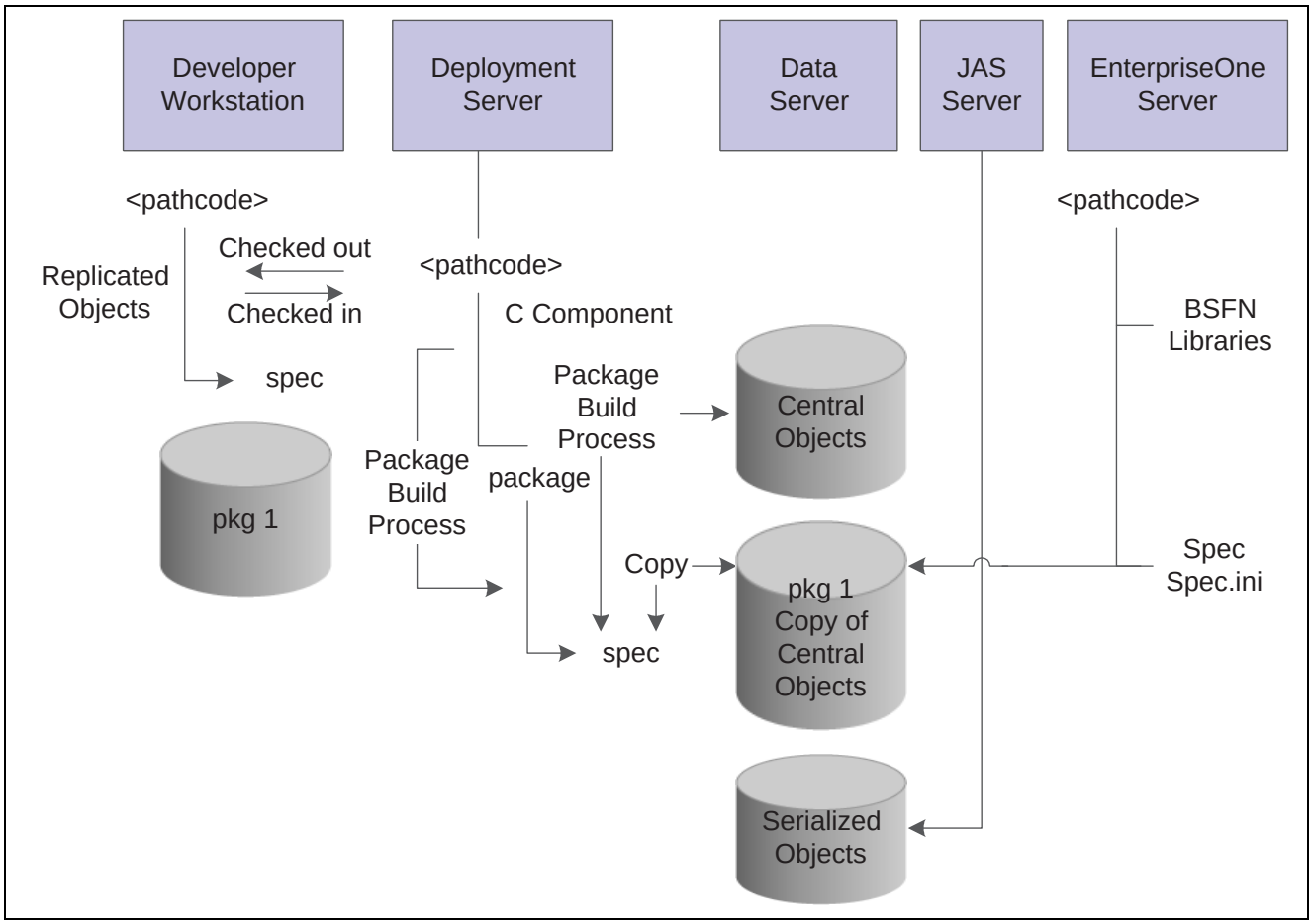
A path code points to a set of objects, therefore a path code definition must associate a set of C components in a directory path with a set of object specifications. In this diagram, you can see how path codes are used to point to both replicated objects on workstations and enterprise servers as well as central objects on the deployment server.

This diagram illustrates the relationship between path codes and object storage:

Path Codes and Object Storage using TAM



Path Codes and Object Storage using TAM

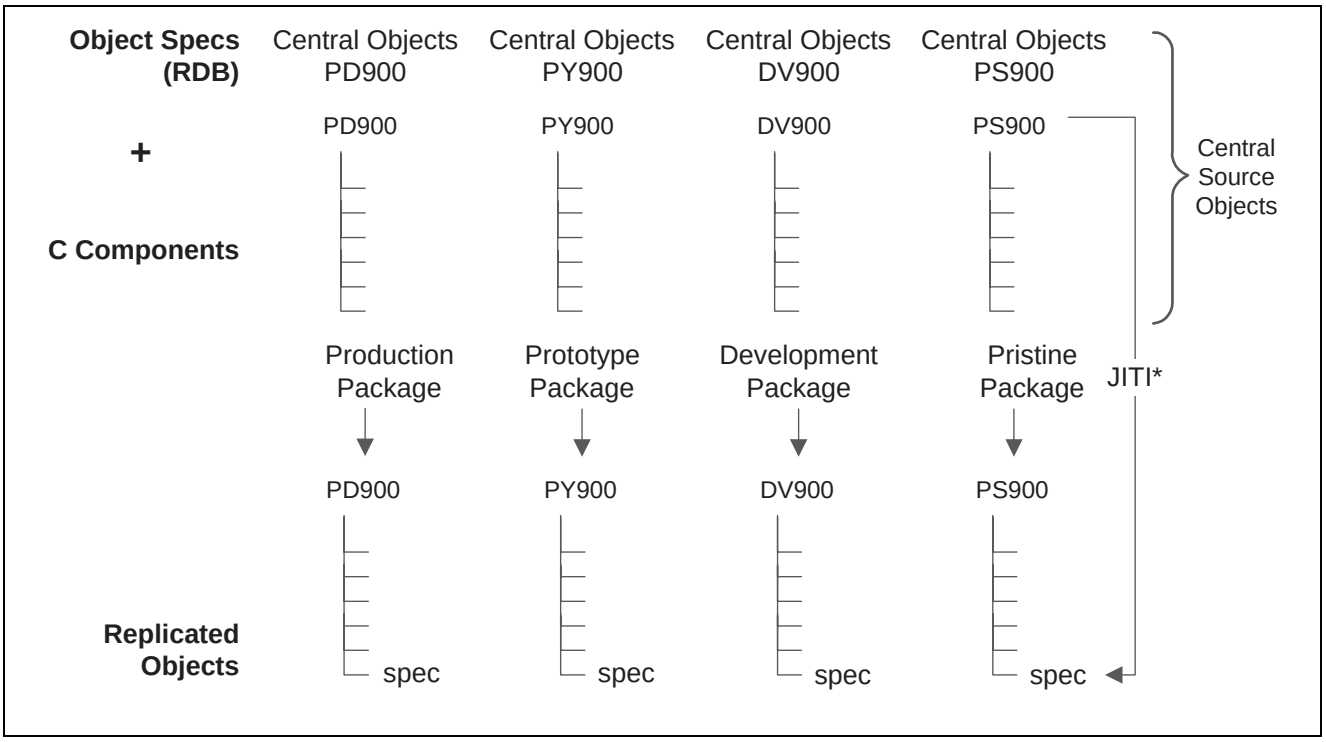


Path Codes and Object Storage using XML

Central Objects and Path Codes

The path code specifies which set of central objects is used when creating a particular package. Once the package has been deployed, the path code also specifies where the new replicated objects reside.

This diagram illustrates the relationship between central objects and path codes:



Central Objects and Path Codes

Setting Up Path Codes

This section lists prerequisites and discusses how to add path codes and create path codes and path code definitions to supplement the one you created during installation.

Understanding Setting Up Path Code

During installation, one path code (PD900) was created for you. The information from this path code can be used to create another one, such as DV900.

When you add a new path code, always copy existing objects, such as from your production path code. You can copy from any existing path code. You cannot add a new path code with an incomplete set of objects.

You must also modify some Object Librarian tables. The Object Path Master File table (F00942) contains all path code definitions for your configuration. This table resides in your system data source.

Note. If you are creating new path codes or creating an environment and a path code, then use the Environment Director discussed in Chapter 9.

CHAPTER 5

Understanding Data Sources

This section provides an overview of data sources and discusses:

- Required data source types
- Database structures
- System data source connections
- System table caching
- Storing object librarian and central objects

Data Sources

The data sources define where the database tables reside and where the software runs logic objects for the enterprise. Data sources can point to:

- A database in a specific location (for example, a local database, such as E1Local located in \E900\data, or an iSeries data library, such as PRODDATA)
- A specific machine in the enterprise that processes logic

Data source definitions are stored in the Data Source Master table (F98611). Workstations use a Common table F98611, which generally resides in the system data source on the enterprise server. Oracle's JD Edwards EnterpriseOne servers that process logic and request data require their own unique definitions for data sources; therefore, they have their own table F98611 in the server map data source.

A least two sets of table F98611 exist. They reside in a centralized system data source normally kept on an enterprise server which is accessed by workstations, and in a server map data source, which each logic server requires.

Data Source Types

Data sources are the building blocks that you use to set up an enterprise configuration. Data sources define all the databases and logic machines required by the Oracle JD Edwards EnterpriseOne configuration. Each database and machine in the enterprise must be defined as a data source for JD Edwards EnterpriseOne to recognize it.

There are two types of data sources:

Database Data Sources A database is a grouping of tables in a database management system. You must identify databases to the applications that access them. You can distribute databases across a network and involve various servers and database management systems. A database data source identifies the database information that the software needs to connect to a database.

Logic Machine Data Sources

A logic machine is the machine on which batch applications and master business functions run. You must identify logic machines using a data source definition. The data source definition must include the network information about the machine, such as a server name - HP9000, for example.

When mapping logic objects for distributed processing, the software uses the machine data source (distributed processing data source) as the target location for processing logic objects.

Data Source Names

Data source names that you define are names used to identify the data source. You should use a meaningful name for the data sources. For example, to indicate that you are storing business data for production users, the data source name could be Business Data - Prod.

JD Edwards EnterpriseOne provides demonstration data source names at installation; you can use these for your own data sources.

See *JD Edwards EnterpriseOne Applications Release 9.0 Upgrade Guide (for your database and platform)*

Data Source Definitions

The data source definition must contain information about the database and the server in which it is located. Different database management systems identify the databases in different ways. For example, you must identify Oracle databases by the Oracle SQL*Net V.2 connect string. You must identify databases that you access through ODBC by the ODBC data source name.

Network Machine/Server Names

Database management systems reside on a machine/server. You must identify this machine/server to the network so that other computers can access its resources. You must provide to JD Edwards EnterpriseOne (in the data source definition) the machine/server name for the machine/server that hosts the database management system in which the database resides.

Required Data Source Types

You must set up a minimum number of data sources for JD Edwards EnterpriseOne to run. Two of the required data sources define machines that process logic in the enterprise. The other data sources define various databases used in the enterprise.

The installation software provides samples of these required data sources to build your system configuration:

Object Librarian

This data source points to the Object Librarian tables you use for custom development. You should have only one set of Object Librarian tables for each software release, regardless of how many path codes (sets of central objects) you maintain. This data source can reside on any supported platform. The Object Librarian data source is named by base release number; for example, Object Librarian - 900.

System

This data source consists of the technical tables you use to run all JD Edwards EnterpriseOne applications. You must set up one system data source per release.

All workstations use a central set of system tables usually stored on the enterprise server but not on the deployment server. Each logic server requires its own subset of system tables. These server system tables are stored in the server map data source.

When running applications, the system tables provide:

- Object mappings (location of tables, batch processes, and business functions)
- Data source definitions
- JD Edwards EnterpriseOne security
- Next IDs (used for development only)

Data Dictionary by Release This data source enables you to store data dictionary master tables in a central location to enable easier administration of changes. Group these master tables together to form a data dictionary database. You should share one data dictionary between the production (such as PD900) and development (such as DV900) path codes. The software allows one data dictionary per path code, but multiple data dictionaries are not recommended or supported. The Data Dictionary data source is named by base release number-for example, Data Dictionary - 900, Data Dictionary - B7334, or Data Dictionary - B732.

Local This data source defines the JD Edwards EnterpriseOne workstation. Use this data source to override the process location of a batch application that you mapped in the Object Configuration Manager to run on the server.

Business Data This data source is used when you divide the business data into multiple owners or libraries, which can reside on the same enterprise server or on different ones. Each group of data requires a separate data source. The installation software provides demonstration data that you can copy to supported host databases. The data source name is Business Data - PS900.

Some examples of business data include:

- Production data (non technical data, such as financial and manufacturing data)
- Test data
- Demo data (demonstration or training data)
- Conference Room Pilot (CRP) data

Distributed Processing This data source definition contains information that the software uses to identify the logic machine in the network. You need to define each logic machine as a data source.

Server Map This data source enables you to create for each logic server its own subset of system tables, which are called server map tables. Server map tables are required for each logic server. You must maintain these tables to ensure integrity with the workstation's system tables.

Use Server Map data sources to establish unique object mappings for logic servers. When batch jobs and business functions running on the server request data, they look to the Object Configuration Master and the Data Source Master tables in the server map data source; this is necessary because the mappings are different.

For example, suppose a user logs on to an environment that maps static local data on the workstation, dynamic transaction data to the server, and the master business functions and batch processes to the server. The user enters a sales order and clicks OK to enter the order, which runs the Sales Order Entry master business function on the server. It does not make sense for the master business function to go back to the workstation to retrieve user defined codes and tax information; therefore, the server map Object Configuration Manager table maps all data to the appropriate server data source.

These tables in the Server Map database are unique to a server's perspective of processing:

- Object Configuration Master (F986101): Provides logic objects processing on a server request data and perhaps other logic objects. When these requests are made to JD Edwards EnterpriseOne running on a server, Object Configuration Master must be accessed to find the correct mappings for the data and logic objects. Servers might have different mapping requirements than workstations.

  For example, you should map all user defined codes locally to the workstation for performance during interactive processing. Server processing would require you to map these files locally to a server database to enhance server processing performance.
- Job Control Status Master (F986110): Records information about batch jobs launched on a server.
- Job Number Master File (F986111): Records next numbers for batch jobs launched on a server.

Central Objects

This data source points to the source objects (central objects specifications), as well as the User Overrides table (F98950). Central Objects data sources are databases.

If you have multiple path codes, each must have a separate Central Objects data source. Developers check objects out of a Central Objects data source for modification. When the developer checks in the objects, the system copies the objects from the developer's workstation to the relational database tables in the Central Objects data source. You must set up one Central Objects data source for every path code needed in the configuration, for example, Central Objects - PD900 or Central Objects - DV900.

You must have a Central Objects data source for:

- Pristine objects
- Production objects
- Development objects

You connect each Central Objects data source to a path code used by the environments that you created for the configuration.

Control Table

This data source consists of user defined codes, menus, and next numbers.

Versions

This data source corresponds to the path code, as in Versions - PD900. It stores versions and processing option information. It includes these tables:

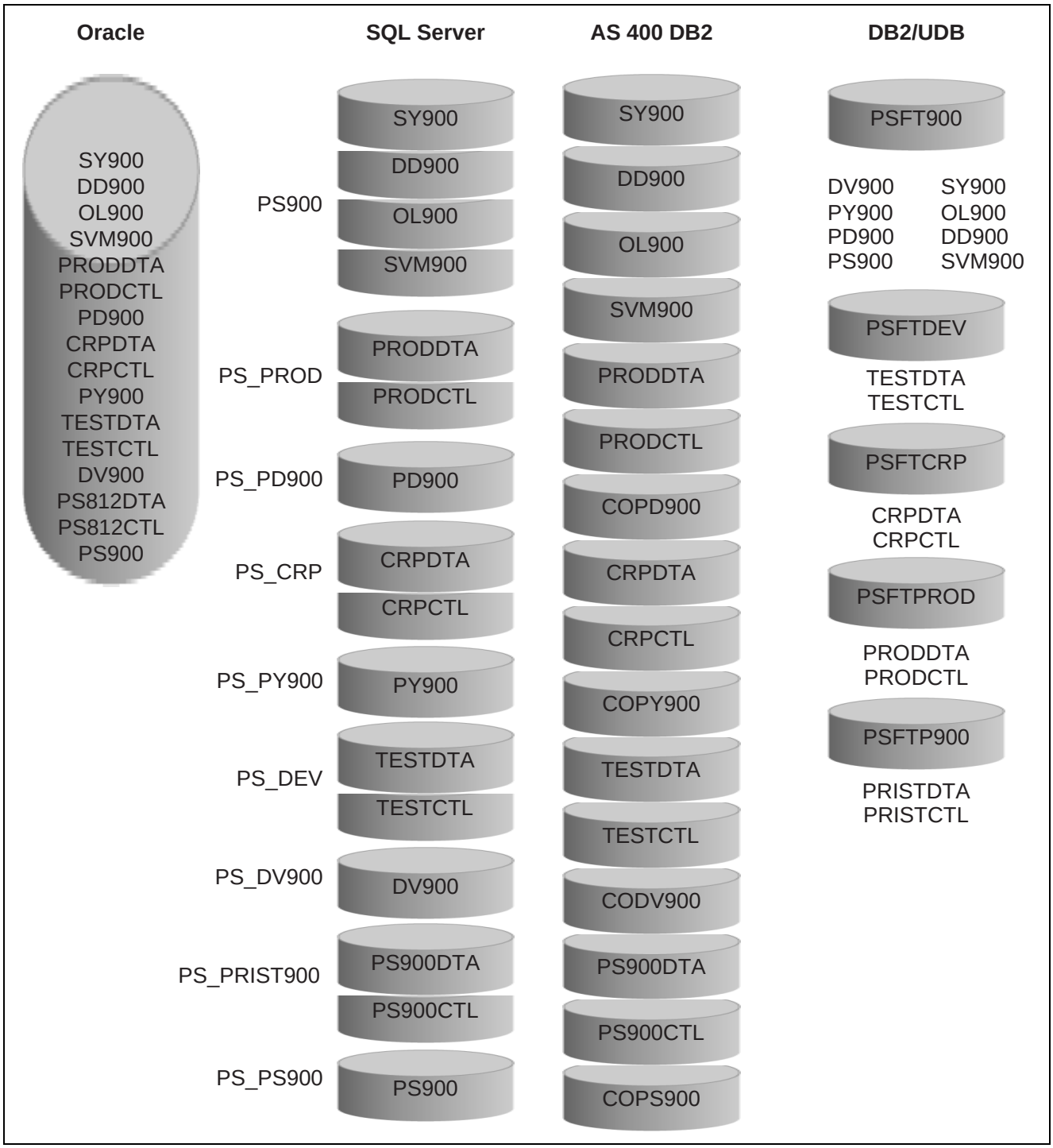
- Versions List (F983051)

- Processing Option Text (F98306)

Database Structures

All supported database platforms have a similar configuration of tables and data sources.

This diagram illustrates owners and databases for four different platforms:



Example of owners and databases structure

Oracle Structure and JD Edwards EnterpriseOne

The basic architecture of an Oracle database includes many different logical and physical storage structures.

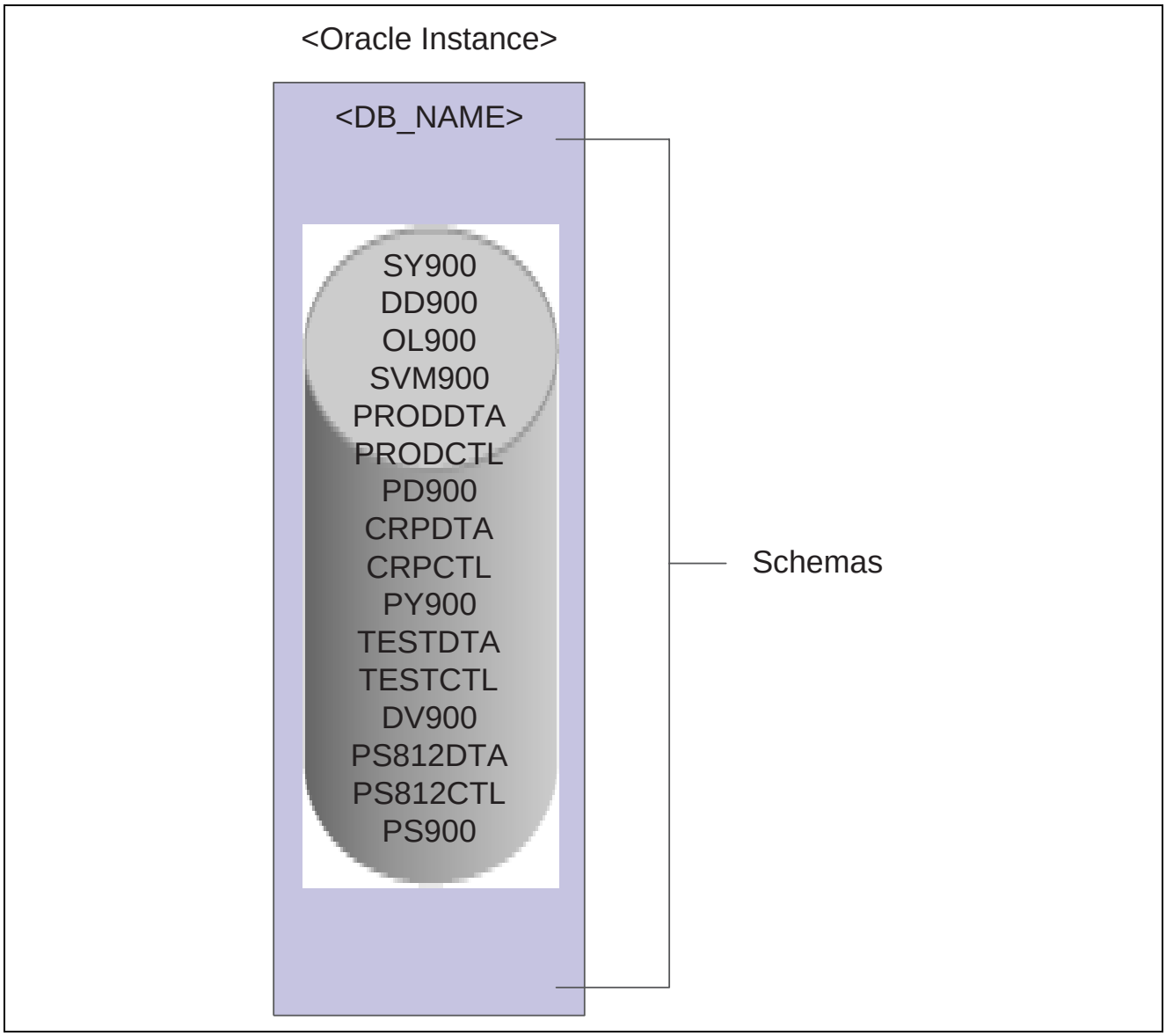
Typically, an Oracle database is divided into one or more logical storage structures. The highest-level structures are table spaces and user schema. These structures provide two categories that data may be logically grouped. Data belonging to one table space may belong to different schema, and data for one schema may belong to different table spaces.

Table Spaces The physical database storage units, data files, are associated with table spaces according to the logical structure of the database. For example, table spaces may be created to separate different categories of data. Table spaces are divided into smaller logical divisions called segments, which are divided further into extents and data blocks. These levels of data storage allow control over how the data files are allocated for physical storage.

User Schema A schema is a set of objects associated with a user. Schema objects include tables and other data structures used by the database. These objects do not directly correspond to data files stored on the server. Each object's data is stored in one or more data files within a table space. You can specify the space allocated for tables and a few other objects.

Tables A schema is a set of objects associated with a user. Schema objects include tables and other data structures used by the database. These objects do not directly correspond to data files stored on the server. Each object's data is stored in one or more data files within a table space. You can specify the space allocated for tables and a few other objects.

This diagram illustrates the Oracle structure with JD Edwards EnterpriseOne:

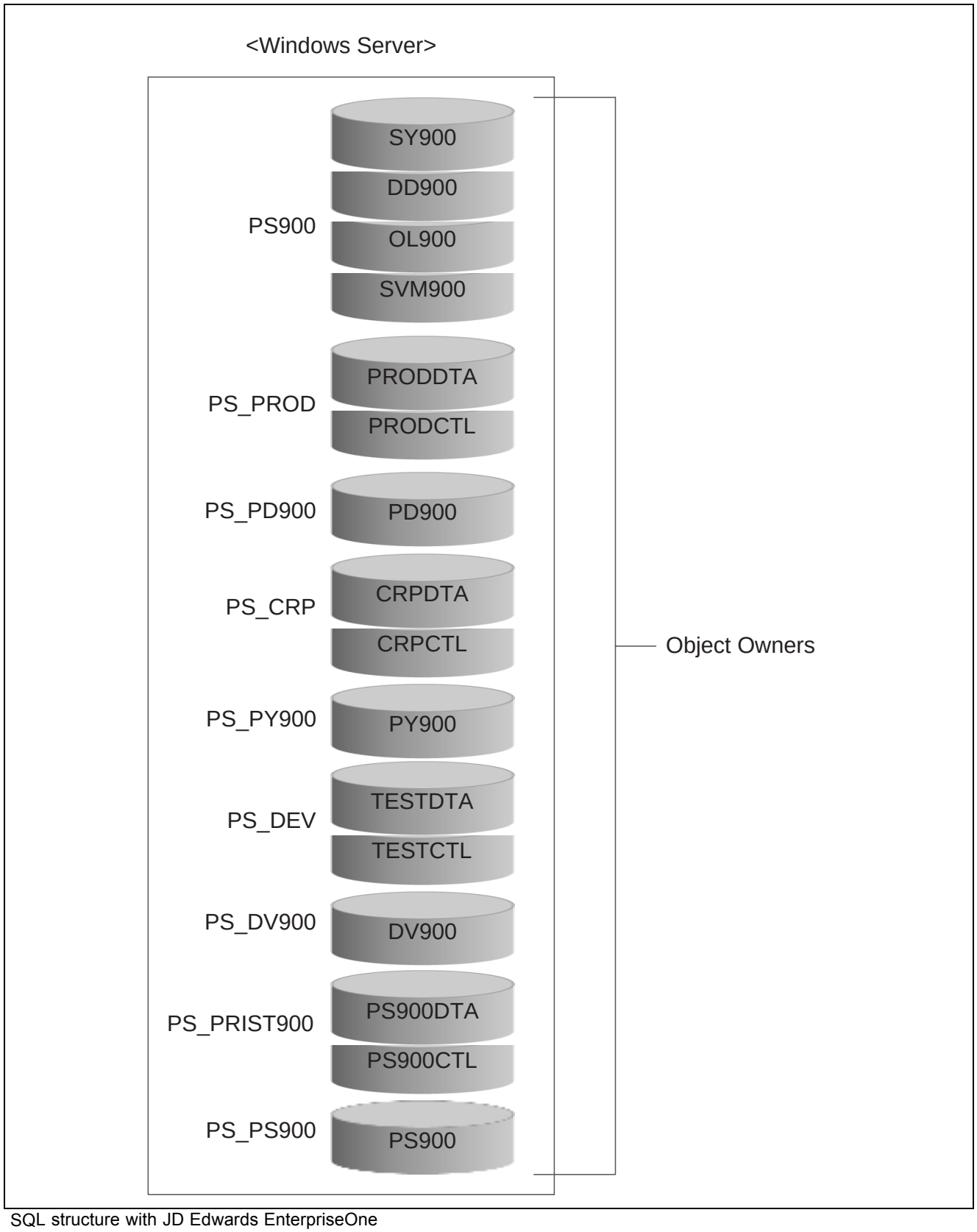


Oracle Structure and JD Edwards EnterpriseOne

SQL Server Structure and JD Edwards EnterpriseOne

SQL Server provides a comprehensive platform that makes it easy to design, build, manage, and use data warehousing solutions which enable your organization to make effective business decisions based on timely and accurate information. SQL Server delivers nine separate databases with JD Edwards EnterpriseOne during an installation.

This diagram illustrates the SQL structure with JD Edwards EnterpriseOne:



SQL structure with JD Edwards EnterpriseOne

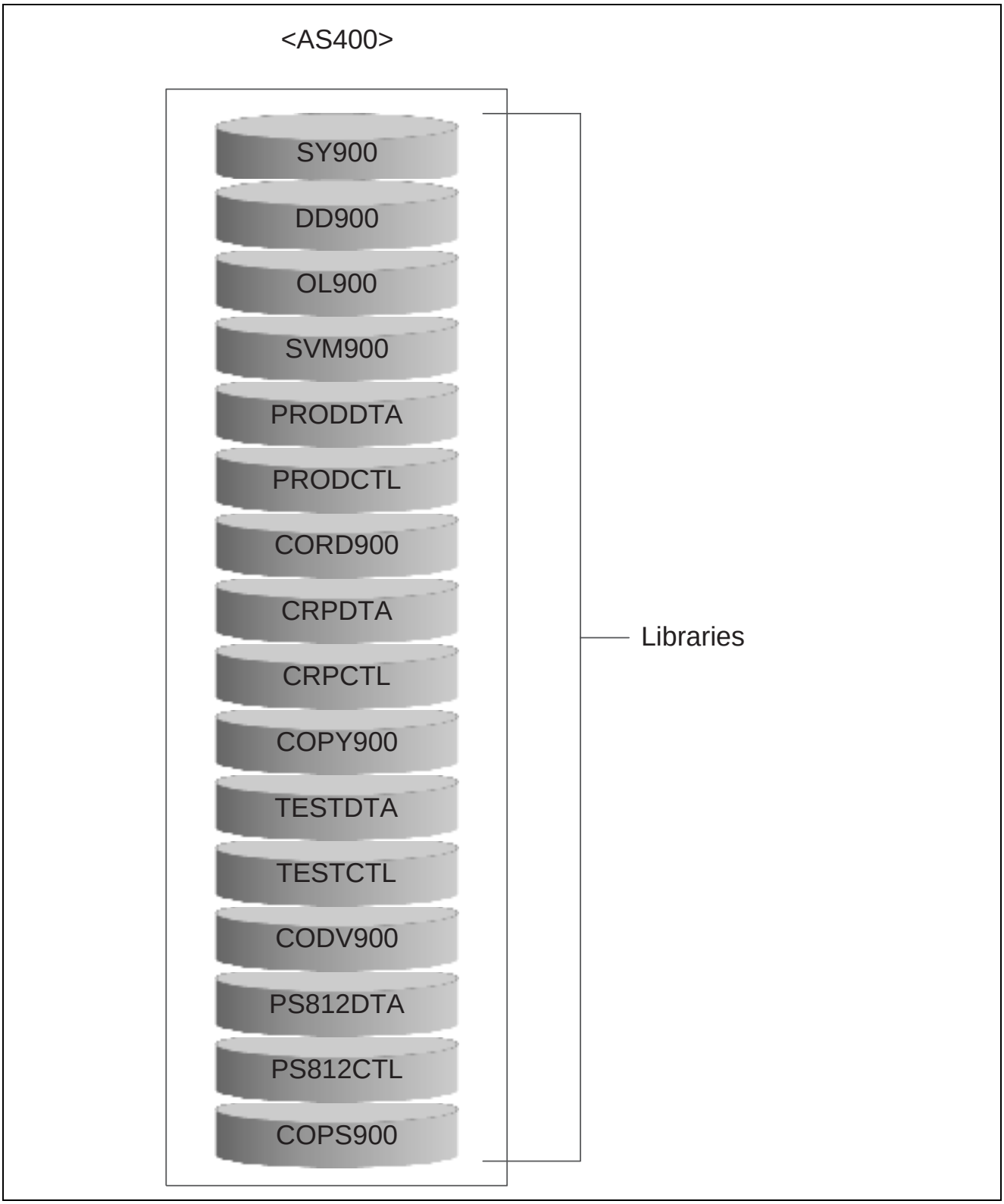
AS400 DB/2 Server Structure and JD Edwards EnterpriseOne

AS/400 DB2 is the relational database manager that is fully integrated and provides numerous functions and features such as triggers, stored procedures, and dynamic bitmapped indexing that serve a wide variety of application types. These applications range from traditional host-based applications to client/server solutions to business intelligence applications.

In the AS/400 system, each file (also called a file object) has a description that describes the file characteristics and how the data associated with the file is organized into records and the fields in the records. The operating system uses this description whenever a file is processed.

AS400 DB/2 installations store all tables in their respective data sources in a single database.

This diagram illustrates the AS400 DB/2 structure with JD Edwards EnterpriseOne:



AS400 DB/2 structure with JD Edwards EnterpriseOne

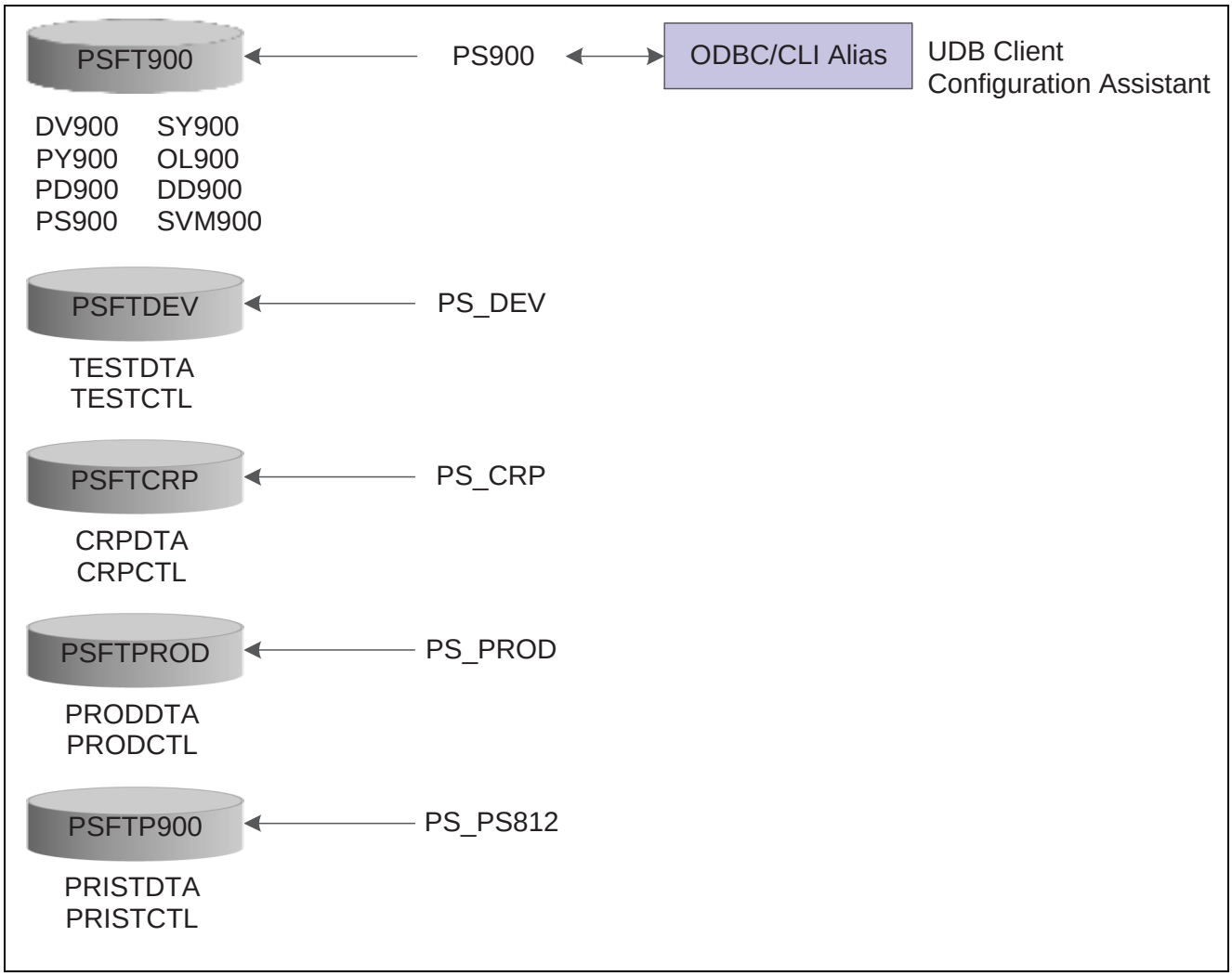
DB2/UDB 8.1.4 Structure for JD Edwards EnterpriseOne

Every data element in a database is stored in a column of a table, and each column is defined to have a data type. The data type places limits on the types of values you can put into the column and the operations you can perform on them. DB2 includes a set of built-in data types with defined characteristics and behaviors: character strings, numerics, datetime values, large objects, nulls, graphic strings, binary strings, and datalinks.

When organizing the data into tables, it is beneficial to group tables and other related objects together. This is done by defining a schema. Information about the schema is kept in the system catalog tables of the database to which you are connected. As other objects are created, they can be placed within this schema.

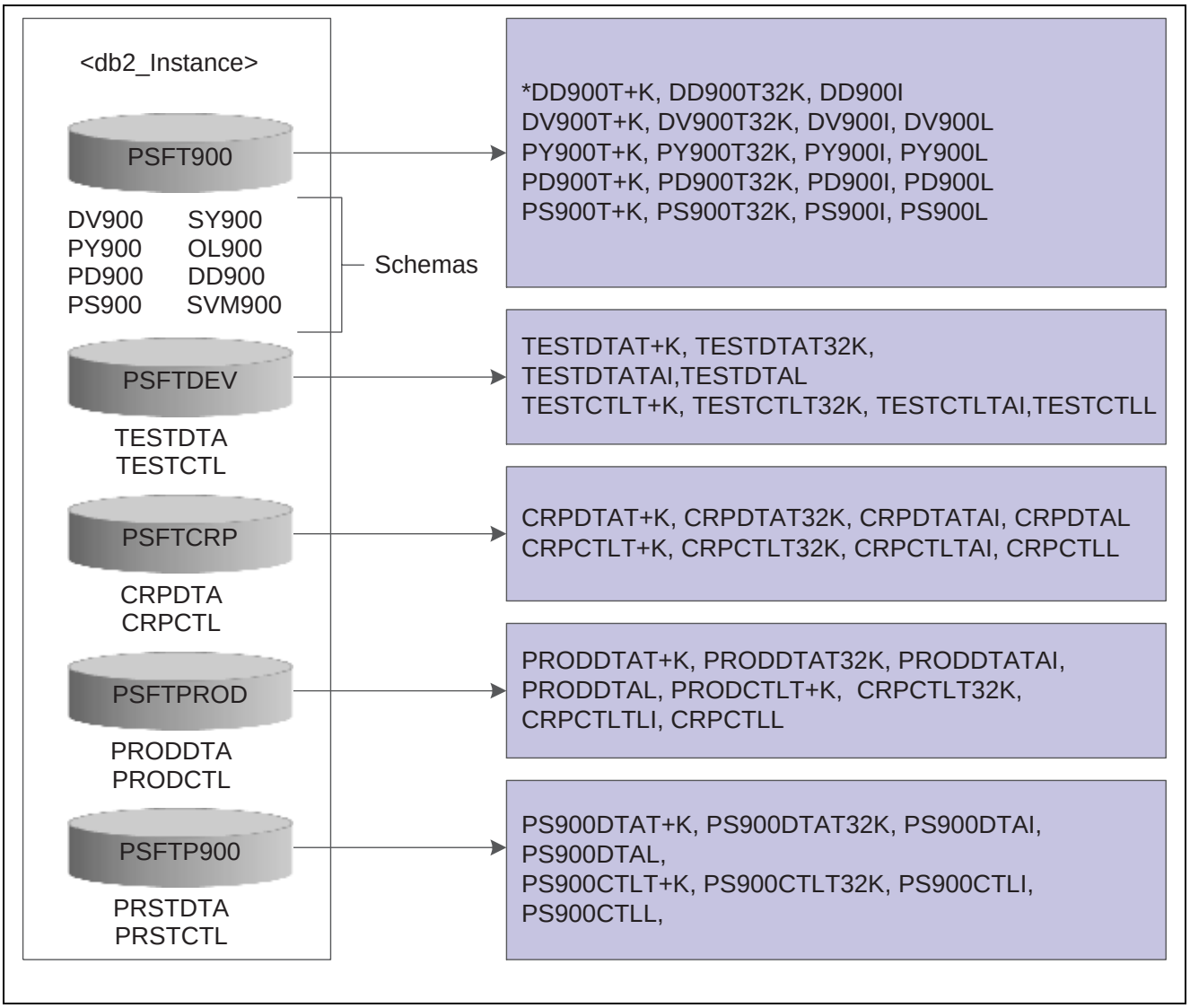
Each schema has a set of four dedicated tablespaces in which the data is physically stored. IBM recommends that each tablespace be stored on a separate disk drive.

This diagram illustrates the DB2/UDB 8.1.4 structure with JD Edwards EnterpriseOne:



Schemas and tablespaces for DB2/UDB 8.1.4

This diagram illustrates the schemas and associated tablespaces for DB2/UDB 8.1.4:



Schemas and tablespaces for DB2/UDB 8.1.4

System Data Source Connections

When JD Edwards EnterpriseOne starts on a workstation, the software attempts to connect to the base data source found in the workstation jde.ini file. If this data source is unavailable, the software attempts to connect to a secondary data source for system information. It is important to have processes for ensuring that the alternate system data source location contains current information. You can maintain an alternate data source's information using table conversion or data replication.

The jde.ini file should look like the example for the primary system data source connection:

[DB SYSTEM SETTINGS]

.

.

Default Env=DEMO900A

Default PathCode=DEMO

Base Datasource=System 900

Database=System 900

.

.

.

Secondary System Data Source connection

[DB SYSTEM SETTINGS - SECONDARY]

Base Datasource=Access32

Object Owner=

Server=

Database=Access32

During installation, the Release Master application relates the system data source to a release. Configuring the release updates the setup.inf file used during the workstation install to create the jde.ini file.

See Also

Major Technical Tables in the *JD Edwards EnterpriseOne Guide 9.0 Installation Supplemental Reference*

System Table Caching

When a user firsts logs on, the software uses the user ID and environment to retrieve information from the system tables for that user and environment. This information is cached in memory on the workstation. Any time a change is made to the central system tables, dynamic caching of the system information occurs for those workstations with an active JD Edwards EnterpriseOne session.

CHAPTER 6

Setting Up Data Sources

This chapter provides an overview of data sources and logic data sources and discusses how to:

- Set up data source
- Add or modify a data source
- Set processing options for database and logical data sources
- Add or modify the Release/Data Source Map table

Understanding Data Source Set Up

The data sources are the building blocks of your configuration, therefore, the proper identification of all required data sources is critical to having a correct configuration to support your business needs.

The database data sources and logical data sources applications enable you add or modify a database data source or logic data source, respectively. After you add a new data source, you must update the Release/Data Source Map table (F00948) using the Release/Data Source Map application.

Understanding Planning for Data Sources

It is important for you to plan for data sources. The proper identification of all required data sources is critical to having a correct configuration that supports the needs of the business. Be sure to include all machines and all databases in your enterprise.

To plan your data sources:

1. Evaluate the location of the system data.
2. Identify all of the enterprise servers in the configuration that will host logic:
    a. Set up one distributed processing data source for each enterprise logic server.
    b. Set up one Server Map data source for each distributed processing data source.
3. Evaluate how many groups of Oracle's JD Edwards EnterpriseOne objects you require to support the business environment, such as production, pristine, and development. Set up one Central Objects data source for each group of objects.
4. Evaluate the various locations of the business data:
    a. Set up one Business Data data source for each database and library that contains Business Data tables that JD Edwards EnterpriseOne applications use.
    b. Set up one Control Table data source for production and one for testing.

Understanding Setting Up Data Source Definitions

These data source types require just one data source definition for each release:

- Object Librarian
- System
- Data Dictionary
- Local

These data source types might require you to define multiple data sources for each type to support the business requirements:

- Business Data (production, test, CRP, and PS900 demo data)
- Distributed Processing (one per logic server)
- Server Map (one per logic server)
- Central Objects (one per path code)
- Control Tables

Understanding Data Source Naming Conventions

When you add data sources, observe these naming conventions:

- Limit to 30 characters
- Text is case sensitive
- Text is space sensitive

Specific exceptions for the Client Access data source are listed in the Client Access Data Source Name section.

Understanding Client Access Data Source Names

The names of Client Access data sources must conform to these standards:

- Limit to 32 characters
- Begin with an alphabetic character
- Do not use these characters: { } [ ] ( ) ? * = ! @ ;

Note. You must type the data source name before you can use the Client Access ODBC driver to access iSeries data.

Understanding Table Owner (Object Owner ID)

Oracle and SQL Server database management systems have unique table owner IDs for each group of tables. For example, the database that contains the system tables might have an owner such as E900SYS. You must identify the table owner ID for Oracle and SQL Server database management systems.

Adding or Modifying a Database or Logical Data Source

This section lists the prerequisites and discusses how to modify an existing database data source.

Prerequisites

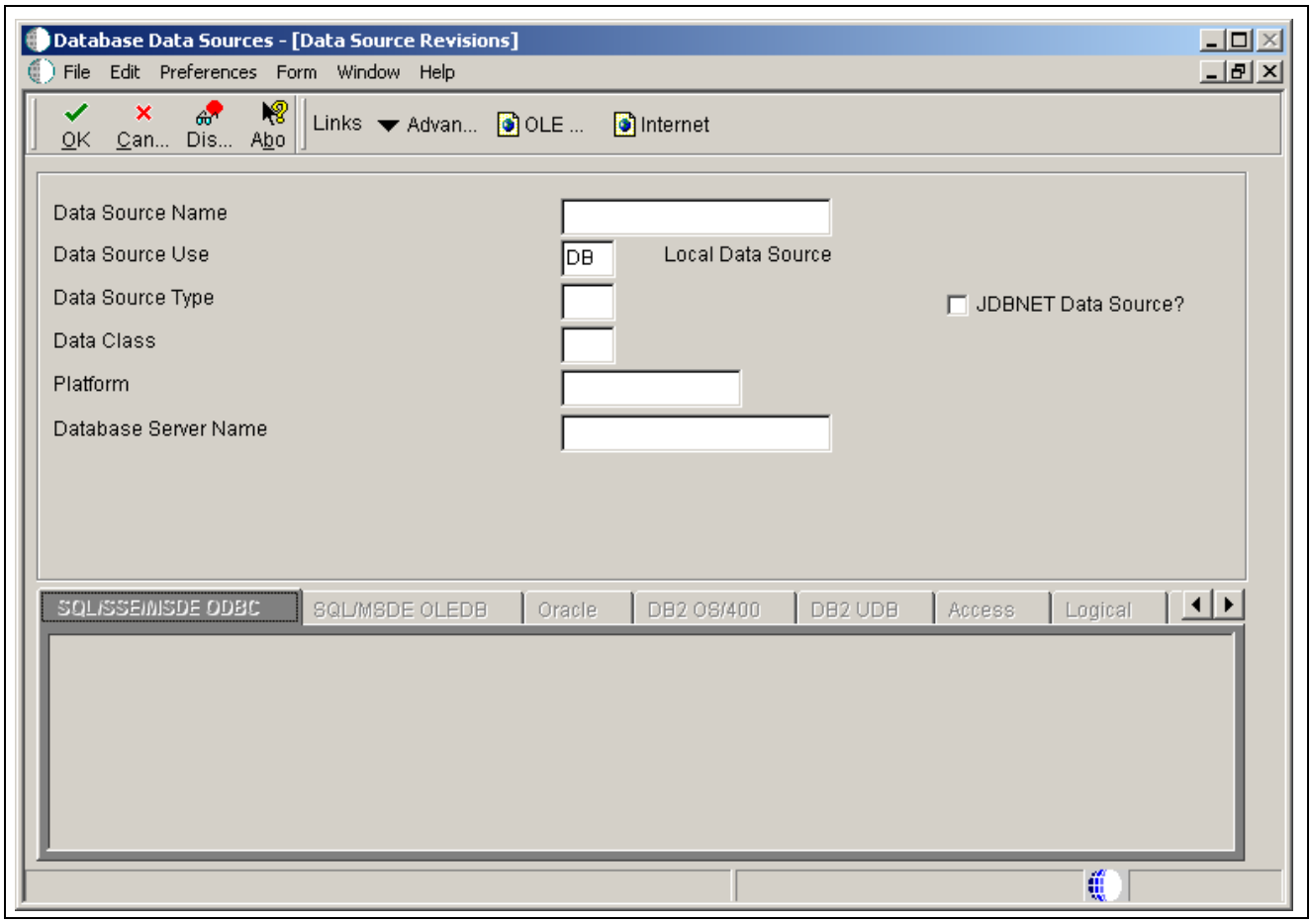
Before you complete the tasks in this section, you should plan for the data sources.

Forms Used to Add or Modify a Database or Logical Data Source

Form Name	FormID	Navigation	Usage
Machine Search & Select	W986115E	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Database Data Sources.	Used to display and select a data source.
Work With Data Sources	W986115A	In the Machine Search & Select form, highlight the machine name and click Select.	Used to add a data source or select a data source to modify.
Data Source Revisions	W986115O	To modify a database data source: In the Work With Data Sources form, click Find, and then highlight the row you want and click Select. To add a database data source: In the Work With Data Sources form, click Add.	Used to add or modify a data source.

Adding or Modifying a Database Data Source

Access the Data Source Revisions form.



Data Source Revisions form

Data Source Revisions Form Fields

Use these processing options to set up the form fields for your data source.

Data Source Name Specify the name that identifies the data source.

Data Source Use Indicate how the data source is to be configured, Servers (SVR) to run UBEs and Business Functions or a Database to access table data.

Data Source Type Specify the type of database.

Data Class Future Use

Platform Specify the type of physical hardware the database resides on.

Database Server Name Specify the name of the computer that receives documents form clients.

SQL ODBC

Use these processing options to set up parameters for a SQL ODBC database.

Object Owner ID Specify the database table prefix or owner.

Database Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

Depending on the data source type, this field is used differently. If the data source type is Oracle, then this field contains the Oracle connect string. If the data source type is Microsoft Access, SQL*Server or Client Access, then this field contains the Windows ODBC data source name. In order to minimize the number of connections to SQL*Server, it is recommended that the ODBC Data Source name is the machine name where the SQL*Server Database resides and that the Catalog name is set for the individual database name. For example, if you have two databases, DatabaseA and DatabaseB, on a machine called INTELNT, this field would contain INTELNT and the Catalog name would be set to DatabaseA for one data source and DatabaseB for the other data source.

ODBC Data Source Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

SQL OLEDB

Use these processing options to set up parameters for a SQLOLEDB database.

Object Owner ID Specify the database table prefix or owner.

Database Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

Depending on the data source type, this field is used differently. If the data source type is Oracle, then this field contains the Oracle connect string. If the data source type is Microsoft Access, SQL*Server or Client Access, then this field contains the Windows ODBC data source name. In order to minimize the number of connections to SQL*Server, it is recommended that the ODBC Data Source name is the machine name where the SQL*Server Database resides and that the Catalog name is set for the individual database name. For example, if you have two databases, DatabaseA and DatabaseB, on a machine called INTELNT, this field would contain INTELNT and the Catalog name would be set to DatabaseA for one data source and DatabaseB for the other data source.

Database Instance Specify the name of the server instance. This name is required for SQL Server ODBC and SQL Server OLEDB data sources.

Oracle

Use these processing options to set up parameters for an Oracle database.

Object Owner ID Specify the database table prefix or owner.

Database Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

Depending on the data source type, this field is used differently. If the data source type is Oracle, then this field contains the Oracle connect string. If the data source type is Microsoft Access, SQL*Server or Client Access, then this field contains the Windows ODBC data source name. In order to minimize the number of connections to SQL*Server, it is recommended that the ODBC Data Source name is the machine name where the SQL*Server Database resides and that the Catalog name is set for the individual database name. For example, if you have two databases, DatabaseA and DatabaseB, on a machine called INTELNT, this field would contain INTELNT and the Catalog name would be set to DatabaseA for one data source and DatabaseB for the other data source.

DB2 OS400

Use these options to set up parameters for a DB2 OS400 database.

Library name Specify the name of the SQL Server database name. This name is required for SQL Server ODBC and SQL Server OLEDB data sources.

ODBC Data Source Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

DB2 UDB

Use these options to set up parameters for a DB2 UDB database.

Schema Specify the database table prefix or owner.

Database Alias Name Specify the name of the SQL Server database. This name is required for SQL Server ODBC and SQL Server OLEDB data sources.

ODBC Data Source Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

Access

Use these options to set up parameters for an Access database.

ODBC Data Source Name Specify the name assigned to the database during installation.

Logical

Server Map Data Source Name Specify the name assigned to the database during installation, such as HPDEVORAP or HP9000.

Setting Processing Options for Database or Logical Data Sources

These options enable you to specify the default processing for programs and reports.

Defaults

Use these options to specify a local (database) data source or a server data source and to indicate whether you want to create an ODBC data source when the Workstation Installation program (P986115) runs.

1. Data Source Type Indicate how the data source is to be configured: Servers (SVR) to run UBEs and Business Functions or a Database (DB) to access table data.

2. ODBC Data Sources Specify whether to create an ODBC data source during the workstation installation process. Values are:

Blank: Do not create ODBC data sources.

1 Create Client Access, SQL Server and DB/2 data sources.

Modifying the Release/Data Source Map Table

This sections provides an overview of the Release/Data Source Map table and discusses how to modify the Release/Data Source Map table.

Understanding the Release/Data Source Map Table

The Release/Data Source Map program (P00948) enables you to define the release level for a data source for a data source type. When you add a new data source, you need to update the Release/Data Source Map table (F00948). This table contains entries for the data dictionary, Object Librarian, versions lists, and workflow, and organizes the entries according to release.

When you copy or transfer objects between data sources, the software checks the release of each data source and then determines whether the source data source and the target data source reside in compatible releases. If Unicode conversion is needed (for example, transfer from an Xe version to a JD Edwards EnterpriseOne 8.10 version), the software converts the data for you when you are using a release of JD Edwards EnterpriseOne 8.10 or greater.

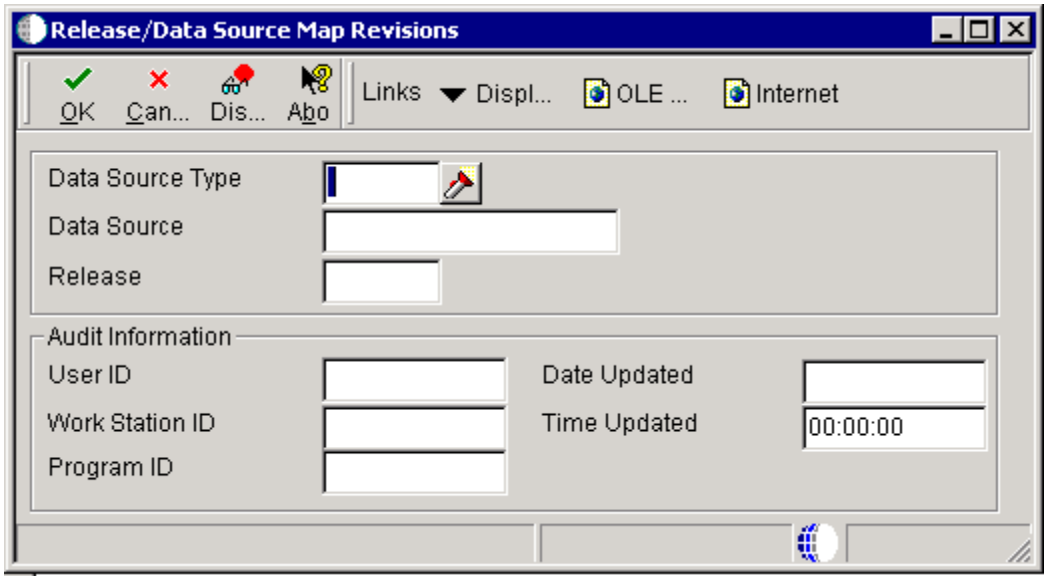
The software does not allow you to move or copy objects between data sources with incompatible releases. When you add a data source, if a record is already found in the Release/Data Source Map table for the data structure and data type entered, the system displays an error. You can only have one release level defined for a data source per data type.

Form Used to Modify the Release/Data Source Map Table

Form Name	FormID	Navigation	Usage
Work With Release/Data Source Map	W00948A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Release/Data Source Map (P00948)	Used to add database sizing information.
Release/Data Source Map Revisions	W00948B	In the Work With Release/Data Source Map form, click Add.	Used to add database sizing information.

Adding or Modifying the Release/Data Source Map Table

Access the Database Object Sizing form.



Release/Data Source Map Revisions form

Data Source Type Specify this is the type of object contained in the indicated data source.

Data Source Specify the name that identifies the data source

Release Specify the release number identified in the Release Master.

UserID Identify the user profile.

WorkstationID Identify the workstation ID that executed a particular job.

ProgramID Identify the batch or interactive program (batch or interactive object). For example, the number of the Sales Order Entry interactive program is P4210 and the number of the Print Invoices batch process report is R42565.

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

T = The first character of the number is alphabetic and identifies the type, such as P for Program, R for report, and so on. For example, the value P in the number P4210 indicated that the object is a program.

SS = The second and third characters of the number is numeric and identify the system code. For example, the value 42 in the number P4210 indicates that this program belongs to system 42, which is the Sales Order Processing system.

XXX = The remaining characters of the number are numeric and identify a unique program or report. fFor example, the value 10 in the number P4210 indicates that this is the Sales Order Entry Program.

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

Time Updated The time the object was last checked in.

Important! If you want to view the data sources without adding or changing one, ensure that you exit from the Data Source Revisions form and click Cancel instead of OK. When you click OK, the system assumes that you have added or changed a data source, and the existing ODBC drivers might not work correctly. You will have to modify them using the ODBC Data Source Administrator applet accessible from the Control Panel.

Setting Advanced Options for Release/Data Source Map Revisions

The Advanced form displays options that are related to the data source type being configured.

Advanced

Use these form exit options to set up additional options.

Unicode Indicate whether a database supports Unicode. This option is required for all data sources.

Use Decimal Shift Check this option to automatically shift decimals when retrieving or updating data based on specifications in the data dictionary. This field is for JD Edwards EnterpriseOne tables only. Non- JD Edwards EnterpriseOne tables should be in a separate data source with decimal shift turned off. If you bring in a JD Edwards EnterpriseOne table as non-JD Edwards EnterpriseOne table, the system does not recognize the fields in this table that have been decimal shifted. This is not checked in Table Conversion.

Use this flag to SELECT or CLEAR row-level record locking for the data source.

You should have this flag turned ON to help prevent database integrity issues.

JDEBASE middleware uses this flag to determine whether or not to use row-level record locking.

Use Julian Dates Check this option to store dates in a Julian format. Otherwise, dates are stored as defined in the Microsoft Windows Control Panel. The system automatically turns this flag off for non-JD Edwards EnterpriseOne tables.

Support For Updates Use this flag to SELECT or CLEAR row-level record locking for the data source.

You should have this flag turned ON to help prevent database integrity issues.

JDEBASE middleware uses this flag to determine whether or not to use row-level record locking.

OCM Data Source Specify whether the data source will contain an OCM table (F986101). If you enable this flag, the system displays a 1 in the OCM Data Source field on the Work With Data Sources form. You should only set this flag for the system data source and any server map data sources. That is, you should not enable this setting for any database data source or log data sources that are not system or server map data sources.

Use Table Owner Activate use of the Owner ID field.

LOB Data Type Support Indicate whether a database supports a LOB data type. This option is required for all AS400 and Oracle data sources.

Note. If you select *I - DB2 UDB* on OS/400 as the data source type, you can select either LOB Data Type Support or iSeries BLOB Support, but not both.

AS/400 BLOB Support The setting for the AS400 Binary Object (BLOB) Data source indicates whether a data source may contain JD Edwards EnterpriseOne BLOB Tables.

CHAPTER 7

Running Data Source Reports

This section provides and overview of data source reports, list prerequisites and discusses how to:

- Run the Data Source Master and Data Source Master Comparison reports
- Run the Verify OCM report

Understanding Data Source Reports

The Oracle JD Edwards Data Source Master batch processes enable you to create a report for a data source or a comparison report for two data sources.

The Data Source Master batch processes are called from a report driver (R98611B), although other Oracle's JD Edwards EnterpriseOne reports do not require a report driver. You use the R98611B report driver to set processing options for each report. Any data selection for a report must be entered for the specific report, not from the driver. When you finish entering processing options and data selection, you run the report from the driver.

Running the Data Source Master and Data Source Master Compare Reports

This section lists the prerequisites and discusses how to set up and run the Data Source Master and Data Source Master Compare reports.

Prerequisites

Before you run the reports in this section, you should complete these tasks:

- Use the report driver to enter processing options for the reports.
- Set up data selection for the reports before running the reports.

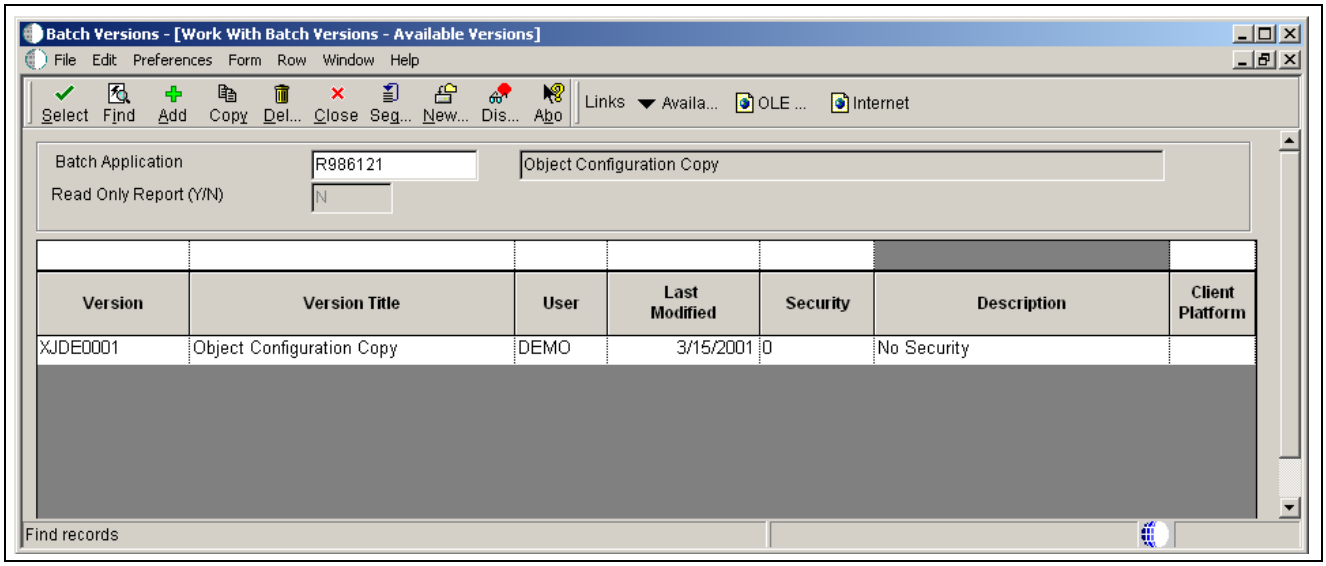
Forms Used to Run Data Source Master and Data Source Master Compare Reports

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Data Source Management Reporting, Data Source Master Report or Data Source Master Comparison (P98305). In the Work With Batch Versions - Available Versions form, enter R98611 (Data Source Master Report) or R986112 (Data Source Master Comparison) in Batch Application.	Used to run the Data Source Master Report or Data Source Master Comparison report.

Running the Data Source Master or Data Source Master Compare Report

The Data Source Master and Data Source Master Compare reports enable you to create a report for a data source.

Access the Work With Batch Versions - Available Versions form.



Work With Batch Versions - Available Versions form

Versions 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 batch jobs or reports. To run a batch process, you must choose a version.
Version Title	A description of the version that appears next to the version number. The version title is different from the report title. This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.
User	Identifies the use ID of the user who last modified the application or version.
Last Modified	Indicates the last time an application or version was modified by the specified user.
Security	This field allows you to restrict user access for a report version. Values are: *0* - No security. Anyone can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is the default when adding a new version. *1* - Medium security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is how JDE Demo versions are delivered. *2* - Medium to full security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version. *3* - Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. *4* - Medium security-extended. Only the "Last Modified By" user can design, change processing values (including runtime processing options and data selection), change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version.
Description	A user defined name or remark.
Client Platform	A category code associated with the Versions List table for JD Edwards EnterpriseOne (F983051).

Setting Processing Options for Data Source Master and Data Source Master Compare Report

Processing options are used to select the report and version you wish to run.

UBE Driver

Use this processing option to enter the report and version that you wish to run. Then complete the processing options on the corresponding tab.

1. Data Source Master Report Name	Specify the name of the report you want to run. Values are: *R98611* - Data Source Master Report *R986112* - Data Source Master Compare

The processing options for the report you are running must be completed. The XJDE0001 version of the actual report is the one that will run. therefore, if Data Selection is necessary, it must be done on the XJDE0001 version of the report.

R986112

Use these processing options to specify the parameters for the R986112 report.

1. Data Source One Specify the names of the data sources that you want to compare. If necessary, you can use the Visual Assist feature to locate valid data sources. Click the Visual Assist button to display the Data Source Search and Select form.

2. Comparison Data Source Two Specify the names of the data sources that you want to compare. If necessary, you can use the Visual Assist feature to locate valid data sources. Click the Visual Assist button to display the Data Source Search and Select form.

3. Comparison Method Specify the comparison method used.

Values are:

1 Compare one direction only. Print only the records found in the first data source, but not in the second data source.

Blank: Compare both directions. Print records found in the first data source but not in the second data source, and also the records in the second data source that don't exist in the first data source.

4. Exceptions Only Specify the type of processing for an event.

Values are:

1 Print exceptions only

Blank: Print all records

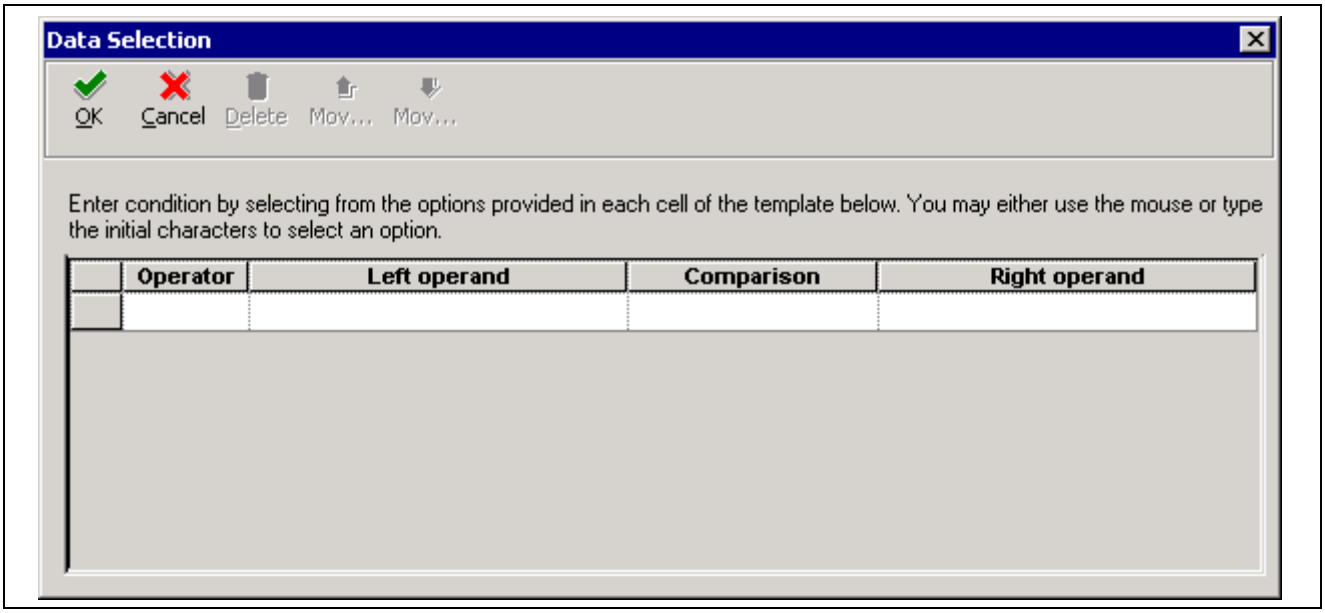
R98611

Use this processing option to specify the data source to use for the R98611 report.

1. Data Source Name Use this processing option to specify the Data Source that you want to use to produce the report.

Setting Data Selection for the Data Source Master or Data Source Master Compare Report

Access the Data Source Selection form.



Data Selection form

Enter a condition by selecting from the options provided in each cell of the template.

Operator	Where
Left Operand	Select the primary data source from the list.
Comparison	is equal to is equal to or empty is greater than is greater than or equal to is less than is less than or equal to is not equal to
Right Operand	Select the secondary data source from the list.

Running the Verify Object Configuration Mappings Report

This section provides an overview of the Verify Object Configuration Mappings report, lists the prerequisites, and discusses how to set up and run the Verify Object Configuration Mappings report.

Understanding the Verify Object Configuration Mappings Report

The Verify OCM report (R9861130) is a useful troubleshooting tool that can help you verify that:

- All Data Source Master definitions have OCM mappings
- Object Configuration Mappings are consistent for a given user, are not duplicated, and have data source definitions,

- Appropriate specification files exist on a specific server
- Business function DLLs exist on a specific server

Run this report to verify the previous items for a workstation or a server. The machine on which you run this report determines which set of Object Configuration Manager and Data Source Master tables will be verified. Consequently, you should run this report on the local machine and on each server that runs JD Edwards EnterpriseOne. You can also run this report on the deployment server.

Prerequisite

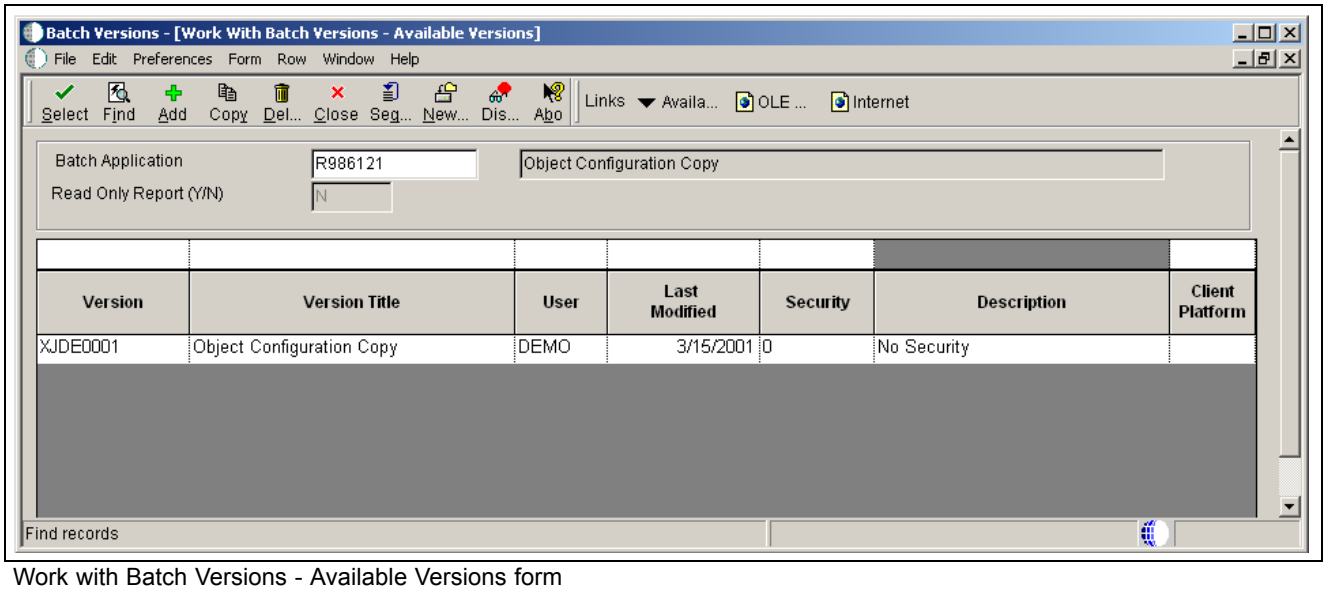
Set processing options for the report from the report driver, not from the actual report.

Forms Used for the Verify Object Configuration Mappings Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Data Source Management Reporting, Verify OCM.(P98305).	Used to run the Verify Object Configuration Mappings report.

Running the Verify Object Configuration Mappings Report

Access the Work with Batch Versions - Available Versions form.



Work with Batch Versions - Available Versions form

Versions 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 batch jobs or reports. To run a batch process, you must choose a version.

Version Title A description of the version that appears next to the version number. The version title is different from the report title.

This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.

User Identifies the use ID of the user who last modified the application or version.

Last Modified Indicates the last time an application or version was modified by the specified user.

Security This field allows you to restrict user access for a report version. Values are:

0 - No security. Anyone can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is the default when adding a new version.

1 - Medium security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is how JDE Demo versions are delivered.

2 - Medium to full security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version.

3 - Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version.

4 - Medium security-extended. Only the "Last Modified By" user can design, change processing values (including runtime processing options and data selection), change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version.

Description A user defined name or remark.

Client Platform A category code associated with the Versions List table for JD Edwards EnterpriseOne (F983051).

Setting Processing Options for the Verify Object Configuration Mappings Report

Processing options enable you to specify the default processing for programs and reports.

Data Source

Use these processing options to specify whether to perform data source validation for data source master definitions, and to specify an environment.

1. Data Source Mappings Specify whether to perform data source validation for data source master definitions. Values are:

Blank: Do not verify data source definitions.

1 Verify that each data source has at least one OCM mapping defined.

2. Environment Validation Specify the environments for which you want to run the validations. Values are:

**All* Run the validations for all environments.

Specific environment name: Validate only that environment.

Parent DLL

Use this processing option to specify whether each business function has a parent DLL assigned.

1. Business Function Parent DLL

Specify whether each business function has a parent DLL assigned. Values are:

Blank: No verification is done.

1 Verify that each business function has a parent DLL assigned. Also, verify that the DLL name is valid.

Specifications

Use this processing option to verify that all required specifications exist for the pathcode.

1. Validate Server Specifications

Verify that all specifications required to run JD Edwards EnterpriseOne exist for the pathcode. This processing option applies only when running against a server. Values are:

Blank: Do not verify specifications

1 Verify whether the required specifications exist

OCM

Use these processing options to specify OCM parameters.

1. User ID to Validate

Specify the user ID for which the OCM validations should be run. Values are:

Blank: Run the validations for all users.

**ALL* Run the validations for all users.

**PUBLIC* Run the validations for only *PUBLIC.

A specific User ID. (Performs validations for only that user.)

2. OCM Mappings

Indicate whether to validate OCM mappings. Values are:

Blank: Do not validate OCM mappings.

1 Validate all active OCM mappings.

2 Validate both the active and inactive OCM mappings.

3. Duplicate OCM Mappings

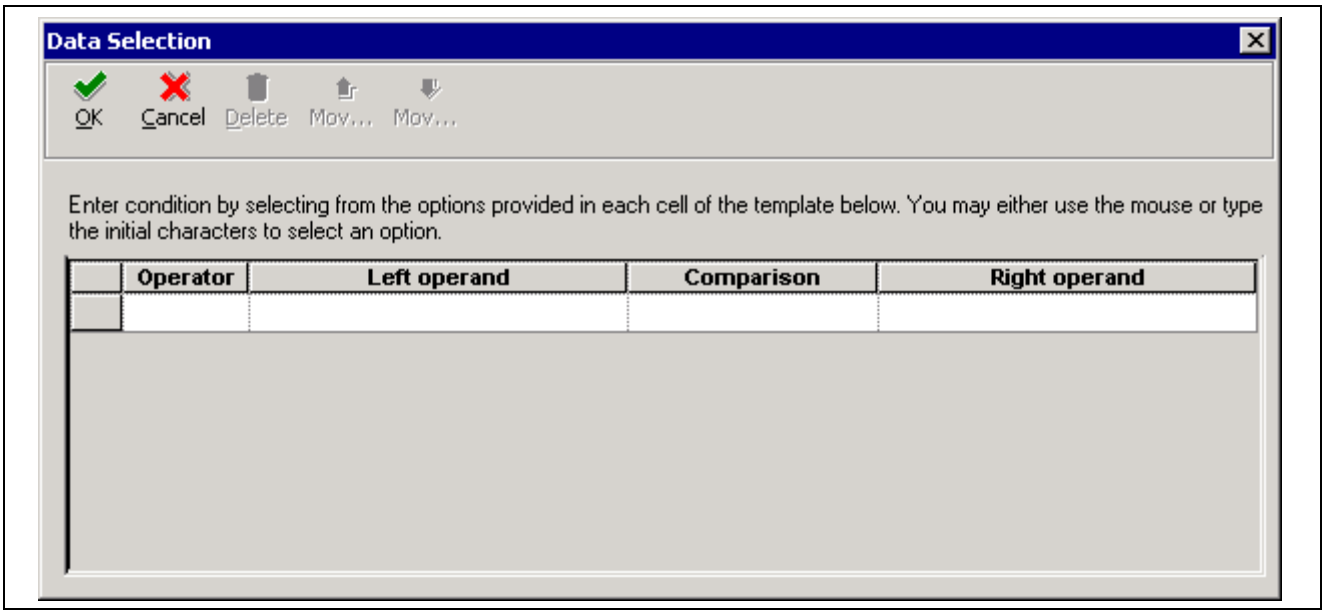
Specify whether OCM records should be checked for duplicates. Values are:

Blank: Do not check for duplicates.

1 Verify that the active OCM mapping records have no duplicates.

Setting Data Selection for the Verify Object Configuration Mappings Report

Access the Data Selection form.



Data Selection form

Enter a condition by selecting from the options provided in each cell of the template.

Operator Where

Left Operand Select the primary data source from the list.

Comparison
- is equal to
- is equal to or empty
- is greater than
- is greater than or equal to
- is less than
- is less than or equal to
- is not equal to

Right Operand Select the secondary data source from the list.

CHAPTER 8

Copying an Environment to a New Environment

This chapter provides overviews of copying an environment to a new environment using Oracle's JD Edwards Environment Director, creating OCM mappings, and UBE copy programs, and discusses how to:

- Set up database components.
- Configure setup files.
- Set Up Security Overrides
- Use Environment Director to copy an environment to a new environment.
- Use Environment Director in the Express Mode.
- Run Environment Director from Different Environments
- Use Object Management Workbench to Modify Table Data Classes
- Use On Track Planning Setup to Modify Table Data Classes

Understanding Copying an Environment to a New Environment

The Environment Director application (P989400) brings all of the steps necessary to create an environment together in one place. It enables the user to copy an existing environment, share or copy an existing path code, set up data sources, create client and server map Object Configuration Manager (OCM) mappings, and copy data from an existing environment. This application works in the same manner as a wizard. You fill in the necessary data and then click Next.

When you want to create a new environment based on an existing environment, you can copy the control tables, business data, central objects, tested full package, and related records from the existing environment to the new environment using Oracle's JD Edwards Environment Director. The copy environment process is run on the Deployment Server.

Note. This process cannot be run for a target environment if the target environment and its path code are already defined in F00941 / F00942 on the Deployment Server. If you want to copy to a pre-defined environment, you must follow the procedure in the Install Guide for DB2 UDB on Microsoft Windows, Chapter 13: Copying an Environment to Another Environment.

See JD Edwards EnterpriseOne Tools Release 9.0 Installation Guide.

Understanding Creating OCM Mappings

You can create OCM mappings (TBLE, GT, BSFN, UBE) for new or copied environments. OCM mappings for a new or copied environment are created by setting up mappings for tables and GTs,and then configuring BSFN and UBE mappings.

TBLE, GT When Creating New Environments

The software finds the data class for each table or GT object in the Object Librarian Master table (F9860). Based on the environment and data class for the object, the data source name is fetched from the data sources by the Environment table (F98511). If a record is not found, the software looks for a record for the data classes where environment name is DEFAULT (a shared data source). If that data source is the default mapping, a mapping with object name DEFAULT is created; otherwise, the actual object name is used. Mappings are not created for tables in the Planner and Internal data classes.

TBLE, GT When Copying Environments

The software looks at each *PUBLIC OCM record for TBLE and GT objects for the environment being copied. It then fetches the data class for the object from Object Librarian. It then looks in F98511 for the data source in the new environment for that data class. If the data source is different from that in OCM, it changes the OCM mapping to point to the new data source.

BSFN, UBE OCM Mappings

The OCM generation algorithm for UBEs and BSFNs is identical, but the output depends on whether the default mapping is LOCAL or an enterprise server.

Default Mapping to LOCAL

LOCAL mappings are created for any object whose Location (labeled Business Function Location for BSFNs and Process Location for UBEs) in OMW is Client Only (F9860.SIBFLOCN = 1).

Default Mapping to Server

Server mappings are created for any object whose Location (labeled Business Function Location for BSFNs and Process Location for UBEs) in OMW is Server Only (F9860.SIBFLOCN = 3).

This table provides the default mappings:

Type	Base Environment (DV900, PS900, PD900, PY900)	JAS/WTS Environment (JDV900, JPS900, JPD900, JPY900)
UBE	Enterprise Server	Enterprise Server
BSFN	LOCAL	Enterprise Server

This table defines the object name and data class.

Object Name	Data Class
F0094	S
F9860	O

Object Name	Data Class
F9200	D
F0101	B
F01012	B
F0004	T
F98710	C
F983051	V
GT92002	D
GT9860A	O
GT3711	B
GT4801	B

This table lists the environment, object name, user role and data source.

Environment	Object Name	User/Role	Data Source
ENV1	DEFAULT	*PUBLIC	Business Data - ENV1
ENV1	F0094	*PUBLIC	System - 900
ENV1	F9860	*PUBLIC	Object Librarian - 900
ENV1	F9200	*PUBLIC	Data Dictionary - 900
ENV1	F0004	*PUBLIC	Control Tables - ENV1
ENV1	F983051	*PUBLIC	Versions - PY900
ENV1	GT92002	*PUBLIC	Data Dictionary - 900
ENV1	GT9860A	*PUBLIC	Object Librarian - 900
ENV2	DEFAULT	*PUBLIC	Business Data - ENV2
ENV2	F0094	*PUBLIC	System - 900
ENV2	F9860	*PUBLIC	Object Librarian - 900
ENV2	F9200	*PUBLIC	Data Dictionary - 900
ENV2	F0004	*PUBLIC	Control Tables - ENV2
ENV2	F983051	*PUBLIC	Versions - DV900

Environment	Object Name	User/Role	Data Source
ENV2	GT92002	*PUBLIC	Data Dictionary - 900
ENV2	GT9860A	*PUBLIC	Object Librarian - 900

Understanding UBE Copy Programs

UBE Copy Programs are used to copy a source package to a target package. Once you have copied the source package to the target package, you must deploy it on the server in order to use it.

Source	UBE Copy Program
Business Data	R98403 XJDE0021
Control Tables	R98403 XJDE0022
Central Objects and Versions	R98403 XJDE0019
Deployment Server Directory	R9800942 (This UBE also copies the Package Header / Detail records)
Checkout Records	R989861 XJDE0001
ESU History	R9672 XJDE0001
Path Code Directory on Enterprise Server	Submits R9894003 XJDE0001 to the server
Copy Metadata Repository	Runs R9894005 XJDE0001 to copy the metadata repository within Central Objects. It is dependent on the existence of the package header record for the target package. R9800942 copies the package records for you.
Configure Manifest	Runs R9894005 XJDE0002 to rename the local metadata repository within the target path / package directory on the Deployment Server. This UBE also configures the manifest in Central Objects and in the target package on the Deployment Server. It also configures the spec.ini in the target package on the Deployment Server. . It is dependent on the existence of the package header record for the target package. R9800942 copies the package records for you.

Setting Up Database Components

This section provides an overview for setting up database components and discusses how to setup database components for the following platforms:

- Unix

- Microsoft Windows
- IBM iSeries

Understanding Setting Up Database Components

Before running the Environment Copy application (P989400), you must set up the database components for the target environment on your enterprise server.

Setting up Database Components for Unix Platform

To set up database components for Unix:

1. Sign on to the enterprise server as an administrative user, for example; root.
2. Change directories: cd to the EnterpriseOne install location, for example; /u01/JDEdwards/e900.

Setting up Database Components for Microsoft Windows Platform

To set up database components for Microsoft Windows sign on to the enterprise server as an administrative user.

Setting up Database Components for IBM iSeries Platform

To set up database components for iSeries:

1. Sign onto the enterprise server as a user with authority to create libraries.
2. Create these libraries:
   - XXXDTA (substitute your business data library)
   - XXXCTL (substitute your control tables library)
   - XX900 (substitute your path code)
   - XX900FA (substitute your target package name)
   - COXX900 (substitute your Central Objects)

Configuring Setup Files

This section provides an overview of configuring setup files and discusses how to:

- Configure Oracle database setup files.
- Configure UDB database setup files.
- Configure SQL database setup files.

Understanding Setup Files Configuration

Before running the Environment Copy application, you must setup the configuration files for the target platform on your enterprise server.

Configuring Oracle Database Setup Files

An Oracle database platform includes:

- Microsoft Windows Platform
- Linux / Unix Platform

Microsoft Windows Platform

To configure the setup file for Microsoft Windows:

1. Navigate to the ORCL folder under the install path.
2. Find the ce_ORCL_SET.bat file and change the variable values:
   - @set NEWENV=YES
   - @set NEWPATHCODE=YES
   - @set UPATHCODE= xx900 (substitute your path code name)
   - @set UENV= xxx (substitute your environment prefix)
3. Start the cmd window, and then navigate to the ORCL folder within the install location (for example, z:\JDEdwards\E900\ORCL).
4. Run ce_InstallOracleDatabase.BAT.
5. Check the logs in the logs directory.

Linux/Unix Platform

To configure the setup file for Linux/Unix:

1. Navigate to the ORCL folder under the install path (for example, /u01/JDEdwards/ORCL).
2. Find the ce_ORCL_SET.sh file and change the variable values:
   - export NEWENV=YES
   - export NEWPATHCODE=YES
   - export LPATHCODE=xx900 (substitute your path code name)
   - export LENV=xxx (substitute your environment prefix)
3. Verify the settings in the ce_CallInstall.sh file: export DATABASE_INSTALL_PATH=/u01/jdedwards/e900/ORCL
4. Run ce_CallInstall.sh.
5. Check the logs in the logs directory.

Configuring UDB Database Setup Files

UDB database platforms include:

- Microsoft Windows Platform
- Linux / Unix Platform

Microsoft Windows Platform

To configure the setup file for Windows with UDB:

1. Navigate to the UDB folder under the install path.
2. Find the ce_UDB_SET.bat file and change the variable values:
   - @set NEWENV=YES
   - @set NEWPATHCODE=YES
   - @set UPATHCODE=XX900 (substitute your path code name - uppercase)
   - @set LPATHCODE=xx900 (substitute your path code name - lowercase)
   - @set LENV=xxx (substitute your environment prefix - lowercase)
   - @set UENV=XXX (substitute your environment prefix - uppercase)
3. Start the db2cmd window, and then navigate to the UDB folder within the install location (for example, z:\JDEdwards\E900\UDB).
4. Run ce_InstallUDBDatabase.BAT.
5. Check the logs in the logs directory.

Linux/Unix Platform

To configure the setup file for Linux/Unix:

1. Navigate to the UDB folder under the install path.
2. Find the ce_UDB_SET.sh file and change the variable values:
   - export NEWENV=YES
   - export NEWPATHCODE=YES
   - export LPATHCODE= xx900 (substitute your path code name - lowercase)
   - export UPATHCODE=XX900 (substitute your path code name - uppercase)
   - export LENV= xxx (substitute your environment prefix - lowercase)
   - export UENV=XXX (substitute your environment prefix - uppercase)
3. Run ce_RunInstallUDBDatabase.sh passing in the current directory as an argument, for example: . ./ce_RunInstallUDBDatabase.sh ‘/u01/JDEdwards/e900/UDB’.
4. Check the logs in the logs directory.

Configuring SQL Database Setup Files

Use these steps to configure the setup files for a SQL database.

1. Navigate to the SQL scripts directory within the EnterpriseOne install location, for example, z:\JDEdwards\E900\MSSQL\scripts.
2. Edit the ce_JDESET.BAT file:
   - rem SQL_PATH - directory where you want your database files
   - @set SQL_PATH= z:\JDEdwards\MSSQL
   - rem Version 8 is SQL Server 2000. version 9 is SQL Svr 2005
   - @SET SQL_VERSION=9
   - @set NEWENV=YES
   - @set NEWPATHCODE=YES

- @set UPATHCODE= xx900 (substitute your path code name)
- @set UENV= xxx (substitute your environment prefix)
- rem JDE_SRV=MYMACHINE or JDE_SRV=MYMACHINE\MYINSTANCE
- @set JDE_SRV=MACHINENAME\INSTANCE (substitute your machine name and instance)

3. Start the cmd window and navigate to the SQL scripts directory.
4. Run ce_InstallSQLDatabase.bat, passing in your sa user and password; for example, ce_InstallSQLDatabase.bat sa MySAPwd
5. Check the logs in the scripts directory.

Setting Up Security Overrides

This section provides an overview of security overrides and discusses these tasks:

- Adding a System User for the Central Objects Data Source Owner.
- Adding an Override for an EnterpriseOne User Running Copy Environment.

Understanding Security Overrides

If you are running with security server turned on, you must add a security override so that the Copy Environment process can create the metadata repository tables in Central Objects. Adding a security override must be done by a security administrator. To add a security override, you must first add a system user for the Central Objects data source owner, and then add an override for the EnterpriseOne user who will run Copy Environment.

Adding a System User for the Central Objects Data Source Owner

To add a system user for the Central Objects data source owner:

1. In P98OWSEC, select the form exit Add System User.
2. On Work with System Users, enter the appropriate data source owner (for example, DV900, PY900 or PD900) in the System User field and select Find.
3. If no values are returned, add the data source owner as a system user by selecting Add and completing the fields on the System User Revisions form:
   - System User
   - Data Source
   - Password
   - Password Verify
4. Click OK and then Cancel and Close to return to the Work with User Security form.

Adding an Override for an EnterpriseOne User Running Copy Environment

To add an override for a JD Edwards EnterpriseOne user who is going to run Copy Environment:

1. On Work with User Security, enter the user who is going to run Copy Environment, and select Find.
2. Select the Form exit, Add Data Source.
3. On Add Data source, complete the User ID, Data Source, and System User fields.

Using Environment Director to Copy an Environment to a New Environment

This section provides an overview of copying an environment to a new environment using Environment Director and discusses how to create a new environment using the Director mode.

Understanding Copying an Environment to a New Environment

The Environment Copy (P989400) application provides a director that steps you through the process of copying an environment to a new environment.

Prerequisites

While Environment Director is a useful tool for creating environments, we recommended using Installation Planner (P9840) and Installation Workbench (P9841) to create custom environments. Environment Director, Installation Planner, and Installation Workbench share common functionality for configuring custom environments.

If the target path code directory already exists on the deployment server or enterprise server (for example, from a previous attempt to copy to that path code), please remove it before you start the copy process. The Copy process breaks if another process has a lock on anything within that directory.

You can use the Windows command:

`rmdir /S /Q d:\JDEdwards\e900\XX900`(substitute your target path code directory)

You can use the Unix command:

```
rm -Rf /u01/jdedwards/e900/XX900
```

On iSeries, if the target path code and package libraries already exist from a previous attempt, please delete them before starting the copy process.

Forms Used By Environment Director in Director Mode

Form Name	FormID	Navigation	Usage
Copy Environment / Package	W989400V	From menu GH9611, run the application Environment Copy(P989400). Select Create/copy to New Environment. ClickNext.	Used to copy an existing environment to a new environment.

Form Name	FormID	Navigation	Usage
Environment Director	W989400A	Click Next from the Copy Environment / Package form.	Defines the components of a new environment including environment, path code, enterprise servers, and data sources.
Environment	W989400B	Click Next from the Environment Director form.	Used to select the mode to be used for setup of the new environment. For Copy Environment, fill in the source environment information.
Environment Properties	W989400C	Click Next from the Environment form.	Used to specify the properties for the new or copied environment. **Note.** If an environment is being copied, the properties are presented as defaults on this form.
Path Code	W989400D	Click Next from the Environment Properties form.	Used to specify the Path Code that will be associated with the new environment.
Path Code Properties	W989400E	Click Next from the Path Code form.	Used to specify the properties for the new path code. The properties for the existing path code are used as defaults for the new path code.
Data Source Revisions	W986115O	Click Next from the Path Code Properties form.	Used to select the data source type and to provide specific database information. After filling in the appropriate information, click OK.
Machines	W989400J	Click OK on the Data Source Revisions form, and then click Next from the Path Code Properties form.	Used to select the enterprise servers where the new environment will run. JD Edwards EnterpriseOne will use this information to populate the server map data sources on these servers. Highlight the machine(s) and click the large right arrow.
Database Script Generator	W989400T	Choose your platform and database type and pressNext.	Used to prompt you for the platform and database type.
Data Load	W989400H	Complete the fields and click Next.	Used to select the data loads the system should perform.
Environment Director Revisions	W989400L	Verify the fields and click End.	Used to complete the environment creation actions.

Form Name	FormID	Navigatton	Usage
Table Conversion Merge/Log	W984052A	Filter on conversion type 95.	The application writes history records into F984052. You can see the history by running Table Conversion / Merge Log (application P984052) on menu GH9611. Filter on conversion type 95.

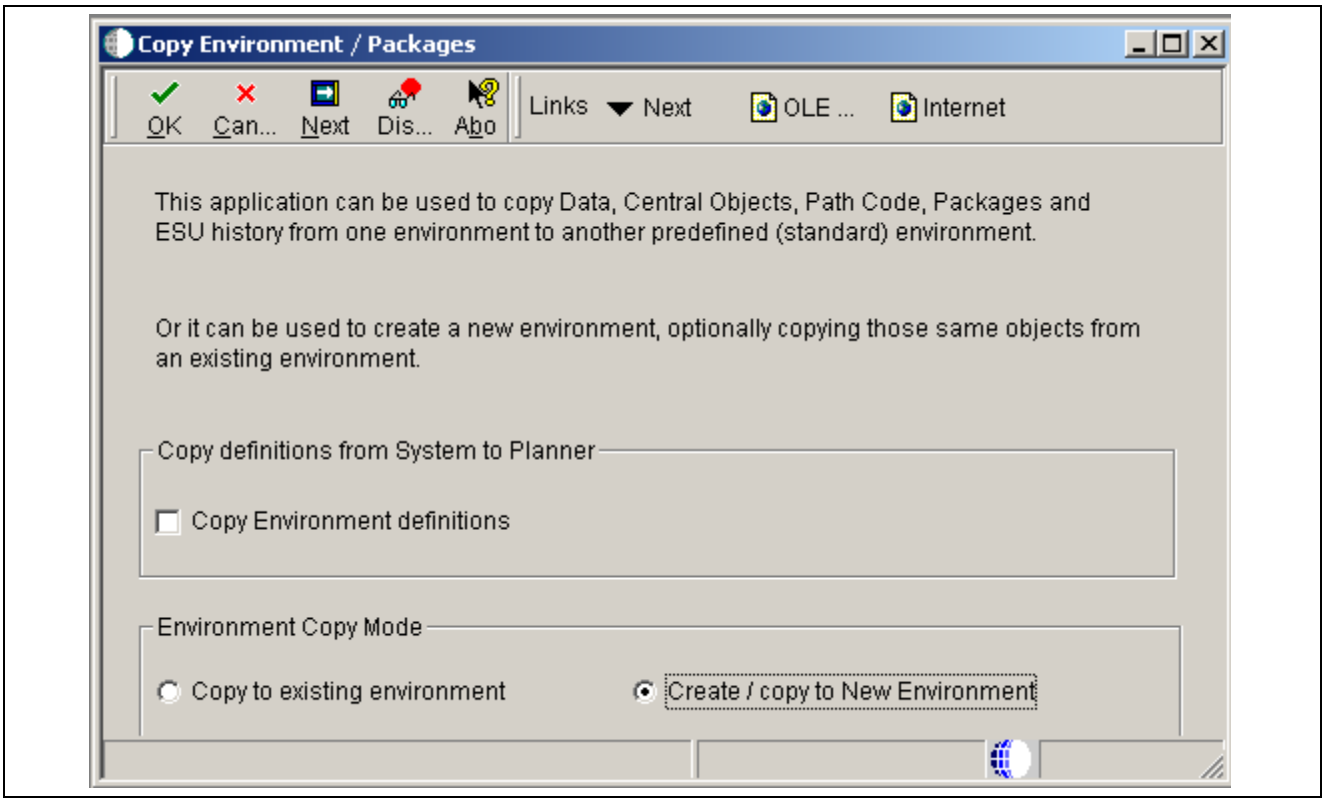
Creating a New Environment Using the Director Mode

Using the Environment Director program (P989400) in Director Mode simplifies creating a new system environment by copying an existing system environment.

Environment Copy

The Copy Environment / Packages form can be used to copy data, central objects, path code, and ESU history from one environment to another predefined (standard) environment, or it can be used to create a new environment, optionally copying those same objects from an existing environment.

Access the Copy Environment / Packages form.

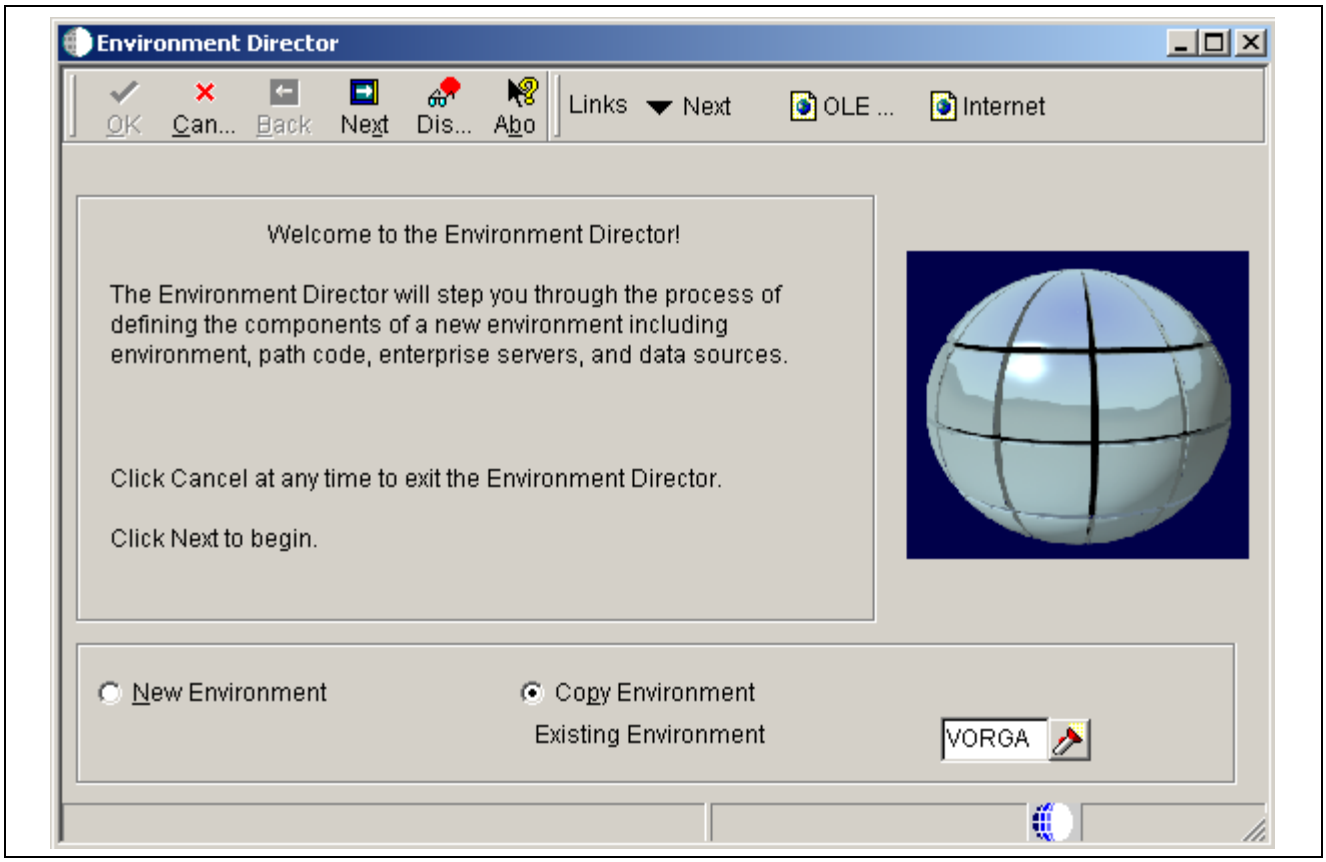


Environment Copy form

Environment Director

The Environment Director steps you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.

Access the Environment Director form.



Environment Director form

New Environment Use this option to create a new environment with no default settings.

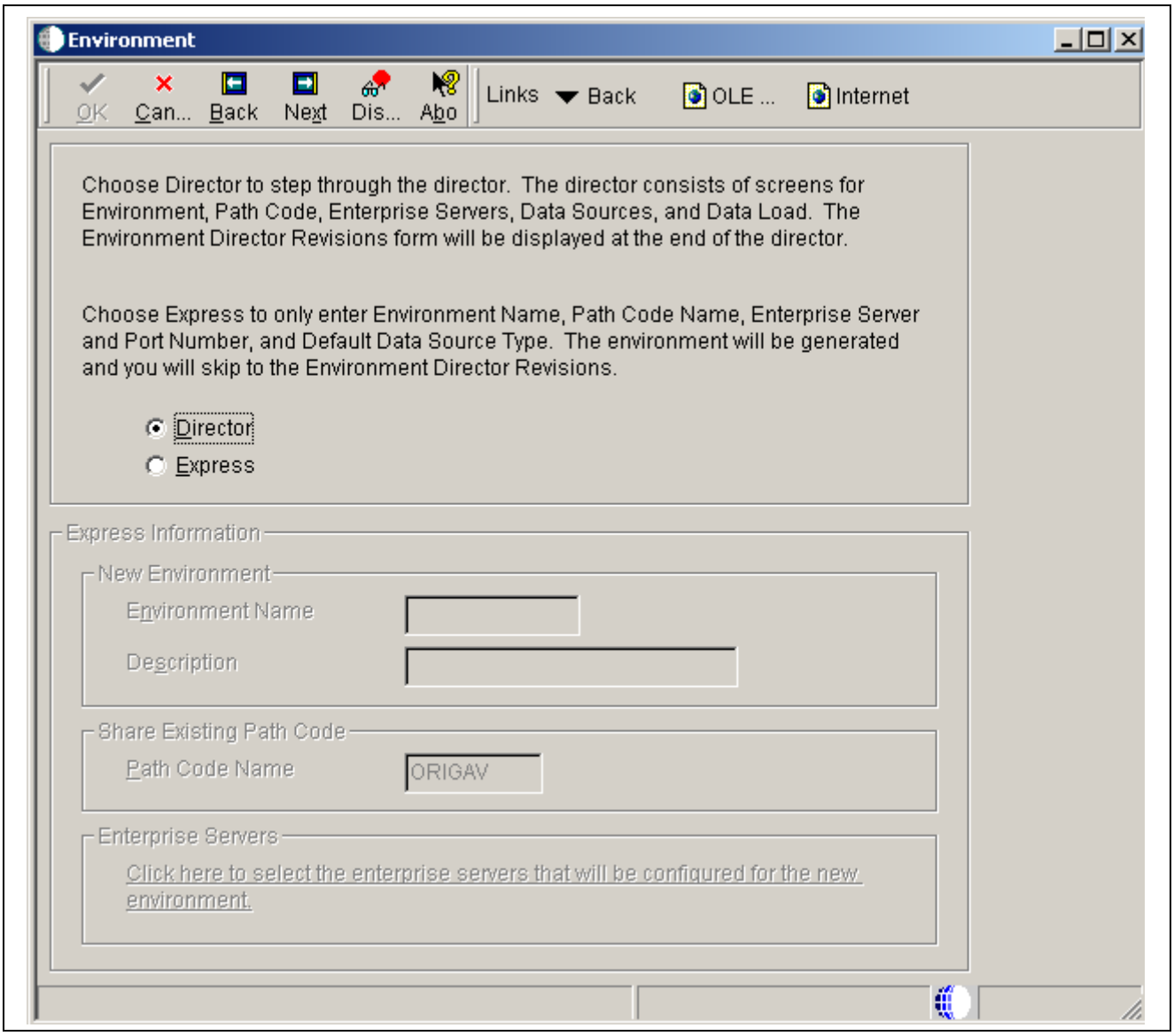
Copy Environment Use this option to use settings from an existing environment as the default settings for the new environment. You can modify these settings if needed.

Note. If you selected Copy Environment, click the search button in the Existing Environment field to select an environment from which you want to copy. The existing environment must be at the current release level as the new environment

Environment

The Environment form is used to select the mode to be used for setup of the new environment.

Access the Environment form and select Director. Press Next to access the Environment Properties form.



Environment form

Director The director mode consists of forms for environment, path code, enterprise servers, data sources, and data load. The Environment Director Revisions form is displayed at the end of the director.

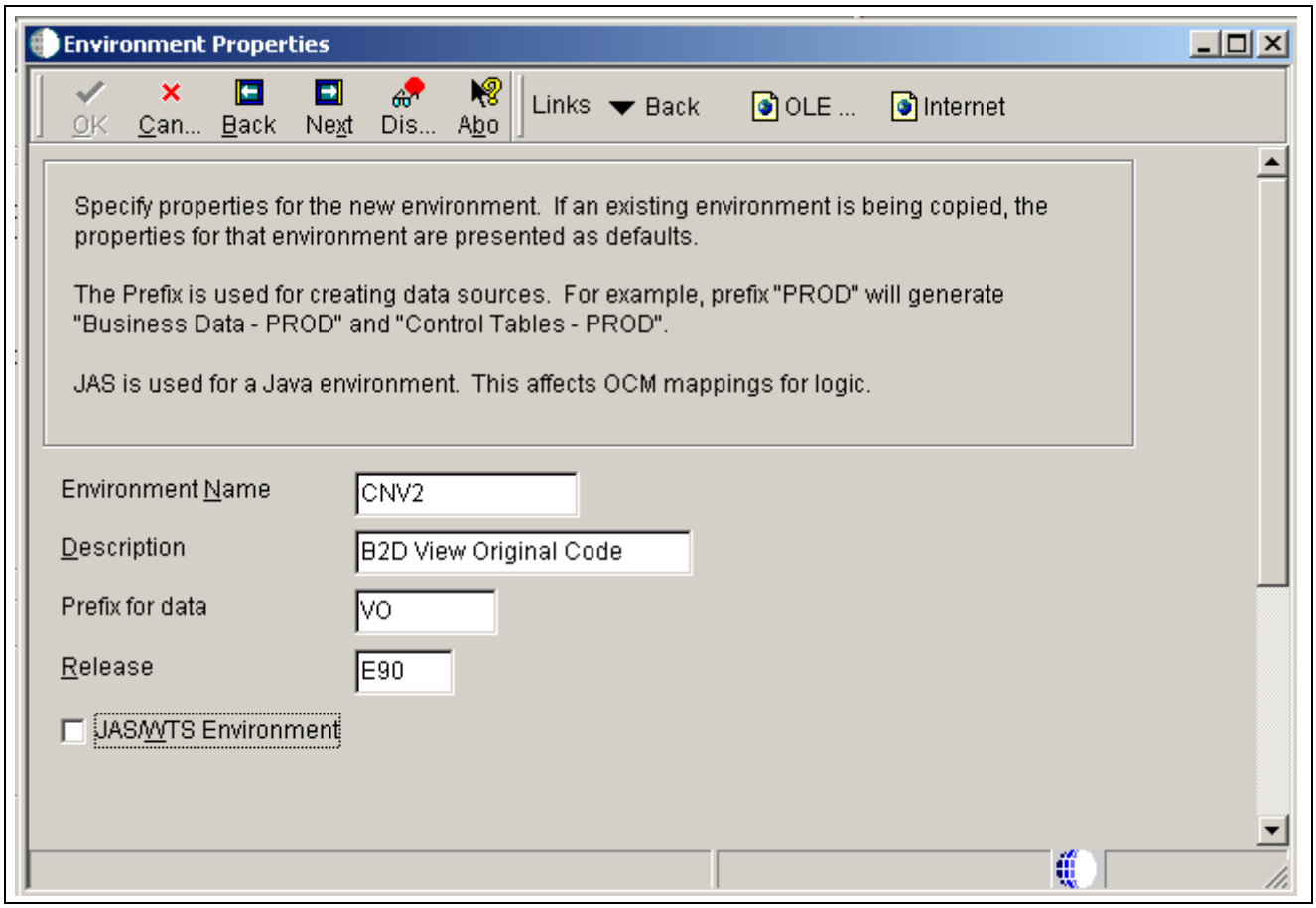
Express The express mode requires that you only enter Environment Name, Path Code Name, Enterprise Server and Port Number, and Default Data Source Type. The environment is generated and you skip to the Environment Director Revisions.

Environment Properties

The Environment Properties form is used to specify the properties for the new environment.

Note. When you copy an environment, the properties are presented as defaults on this form.

Access the Environment Properties form.



Environment Properties form

Name A name for the environment that encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace.

Description A user defined name or remark.

Code A value that the system will substitute for a token when the system creates a new environment from a template.

Release The release number as defined in the Release Master. The default is the current release.

Just In Time Installation This option enables just-in-time installation for anyone signed onto this environment. Consider turning just-in-time installation off before you transfer modified applications into the production path code. After you have fully tested the application and are ready for production users to receive the changes, you can turn just-in-time installation back on.

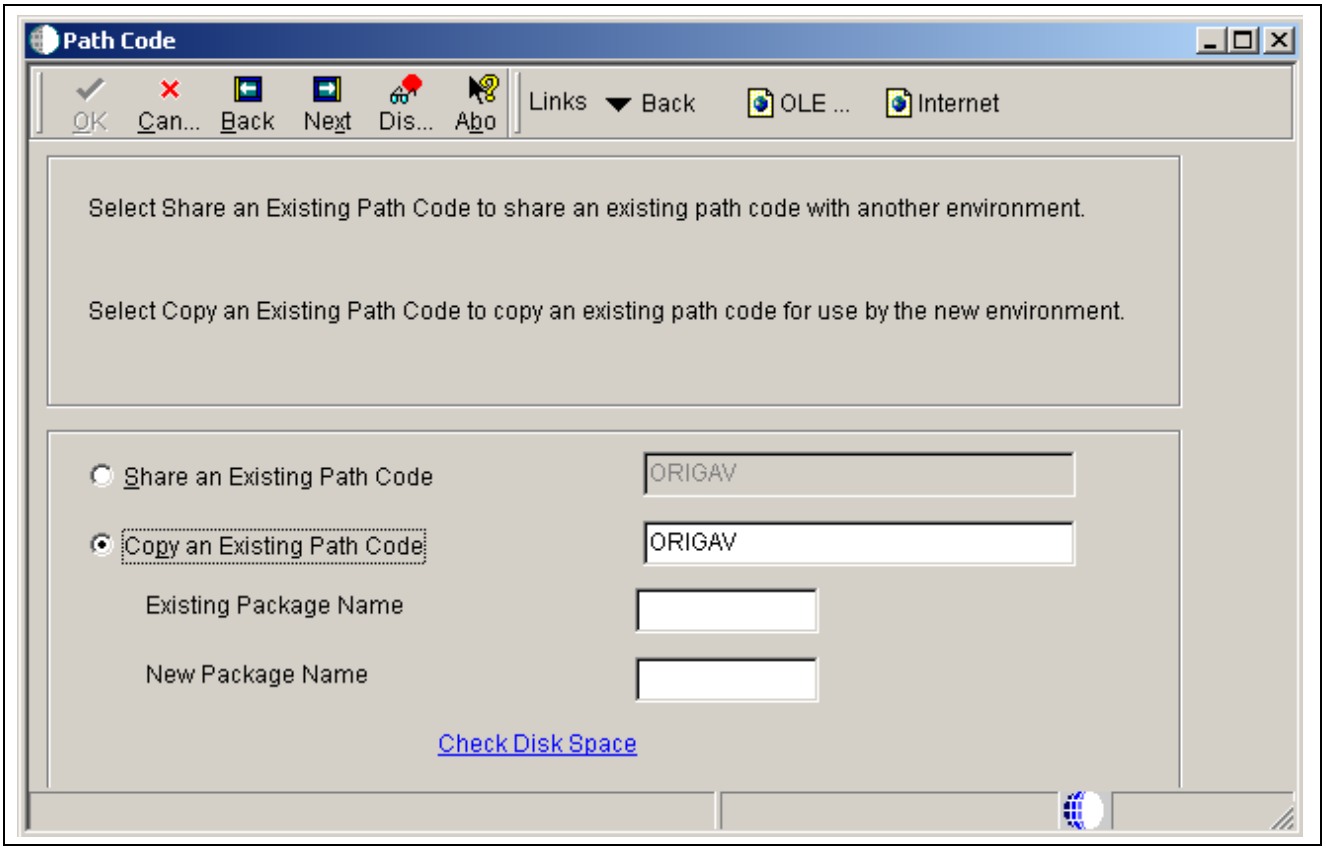
JAS/WTS Environment This option is for a Java or Windows Terminal Server environment. Enabling this option affects how the system creates business function mappings for the environment and which environments are displayed at sign-in.

Path Code

The Path Code form is used to specify the path code that will be associated with the new environment.

To copy the path code, click on Copy an Existing Path Code. To copy a package from that path code, you must choose a full package. The name you choose for your new package must not already exist as a package for any path code. The application fills in the information based on the source path code.

Access the Path Code form.



Path Code form

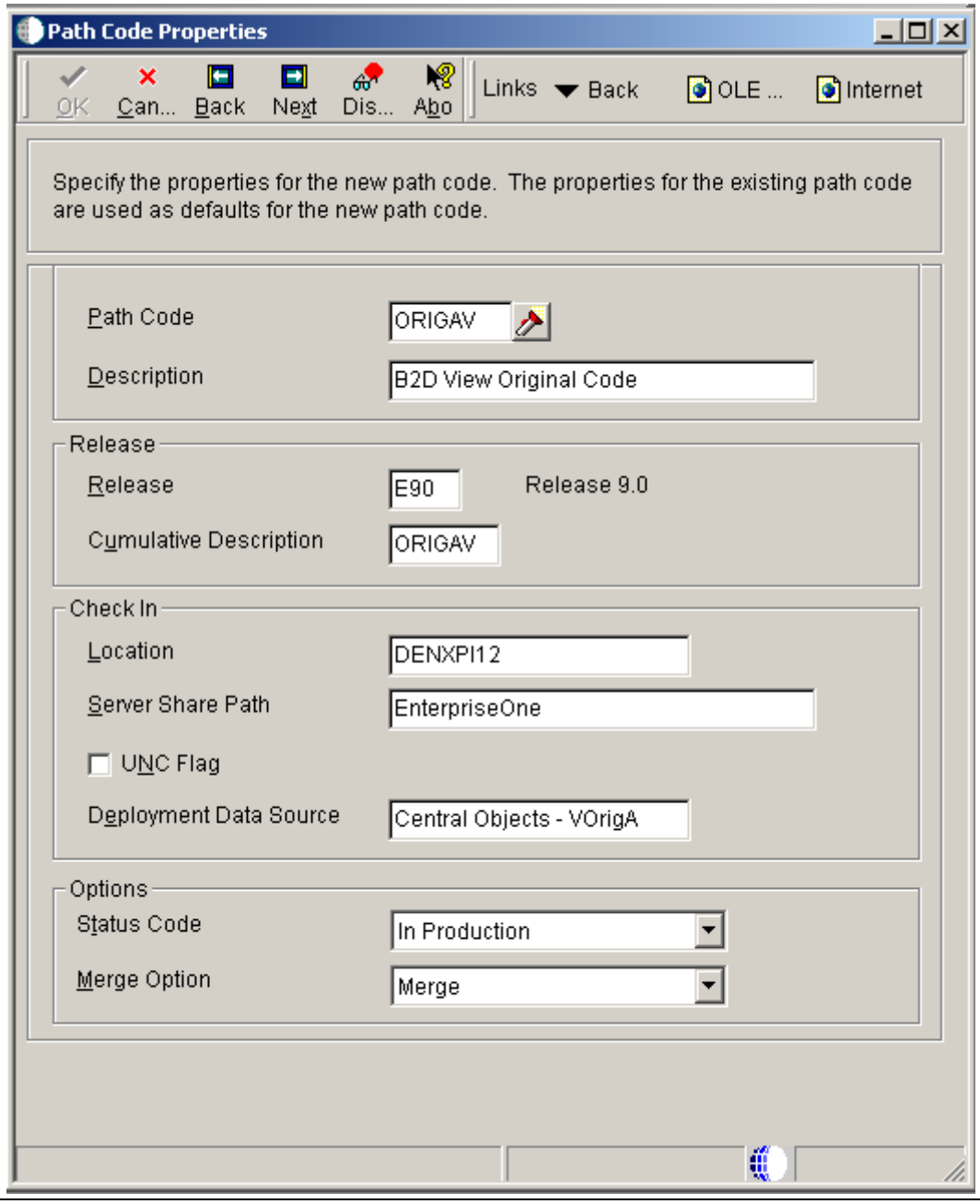
Share an Existing Path Code The path code is a pointer to a set of JD Edwards EnterpriseOne objects and is used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Copy an Existing Path Code The path code is a pointer to a set of JD Edwards EnterpriseOne objects and is used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Path Code Properties

The Path Code Properties form is used to specify the properties for the new path code. The properties for the existing path code are used as defaults for the new path code.

Access the Path Code Properties form.



Path Code Properties form

Path Code Enter the path code, which is a pointer to a set of system objects and is used to keep track of sets of objects and their locations.

Description Enter a user defined name or remark.

Release Enter the release number as defined in the Release Master. It must match the environment's release.

Cumulative Description Use to further identify the release level of the path code; it is only used by OMW for SAR integration

Location Enter the location or machine key of the machine (server or workstation) on the network.

Server Share Path Use this field to indicate the shared directory for this path code. The objects that are stored on a file server will be found in this path.

UNC Flag Determines how to create the server path. Valid options are:

Checked

Creates the path using relative paths. Enter a double slash (\\), rather than the specific followed by a single slash (\).

Unchecked

Creates the path using the actual drive letter.

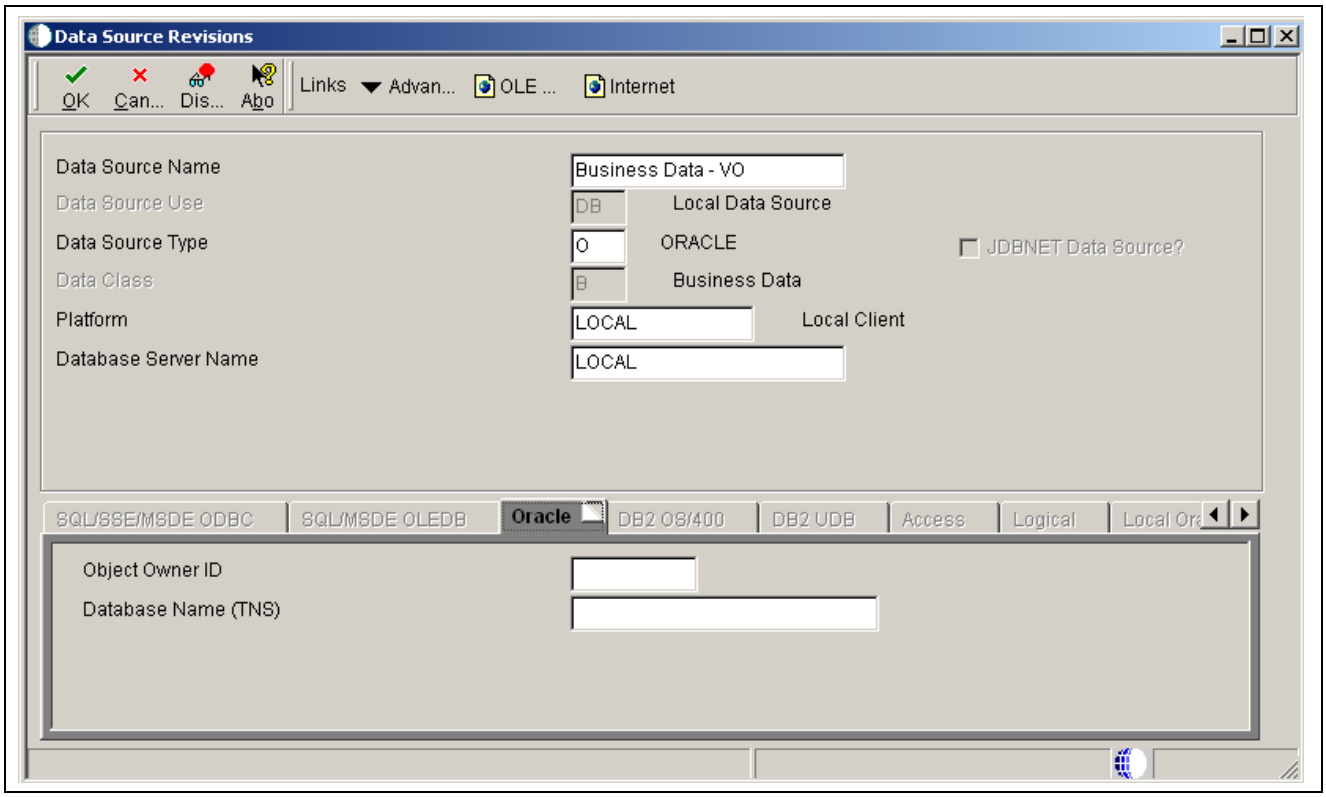
Deployment Data Source Enter the location (data source) of the Central Object Specifications data source that corresponds to the path code. For example, if the environment has a PDxxxx path code, where xxxx is the current system release level, a valid data source for that path code would be Central Objects - PDxxxx, where xxxx is the current release level.

Status Code Enter a code to determine the status of the software in the development cycle.

Merge Option Select this option to determine whether a customer's object will be merged in with the JD Edwards EnterpriseOne object. The merge option can be set at the path code level so that all objects checked into that path will carry the same merge option as the path code.

Data Source Revisions

Access the Data Source Revisions form.



Data Source Revisions form

Data Source Name Specify the name that identifies the data source.

Data Source Use Indicate how the data source is to be configured, Servers (SVR) to run UBEs and Business Functions or a Database to access table data.

Data Source Type Specify the type of database.

Data Class Future Use

Platform Specify the type of physical hardware the database resides on.

Database Server Name Specify the name of the computer that receives documents form clients.

Machines

The Machines form is used to select the enterprise servers where the copied environment will run. JD Edwards EnterpriseOne will use this information to populate the server map data sources on these servers.

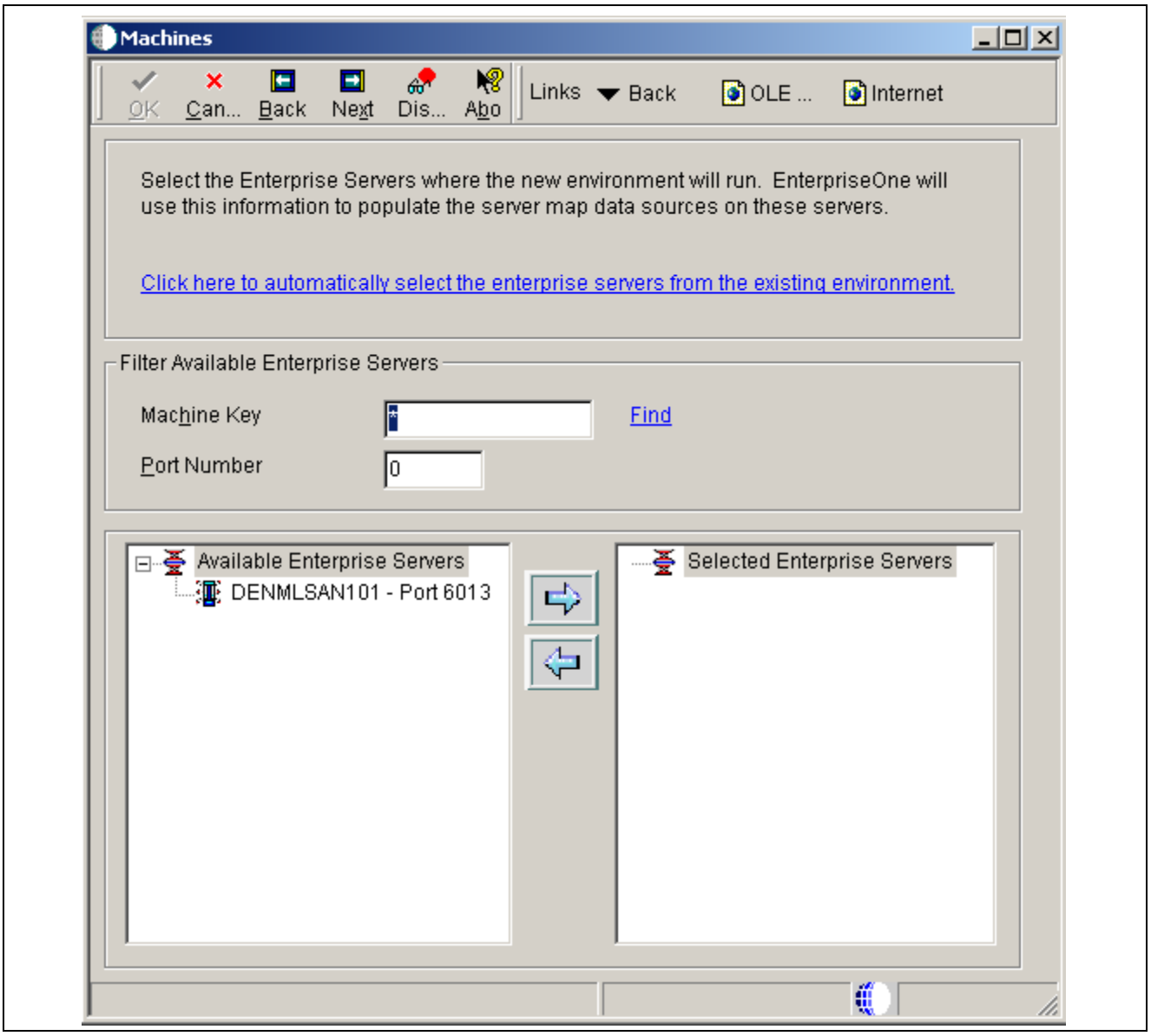
Select which enterprise servers you want to copy the package / path codes on. The existing (source) path code and package must exist on any server you select. If you wish to copy path and package directories form one server to another, you must do the copy yourself using operating system utilities.

For example:

- iSeries – use SAV and RST to get the IFS directories over
- Windows – map a network drive and copy the directories
- Unix – use tar, zcat and ftp to get the paths across machine

You are responsible for ensuring that the binaries are compatible between the source and target machines.

Access the Machines form.



Machines form

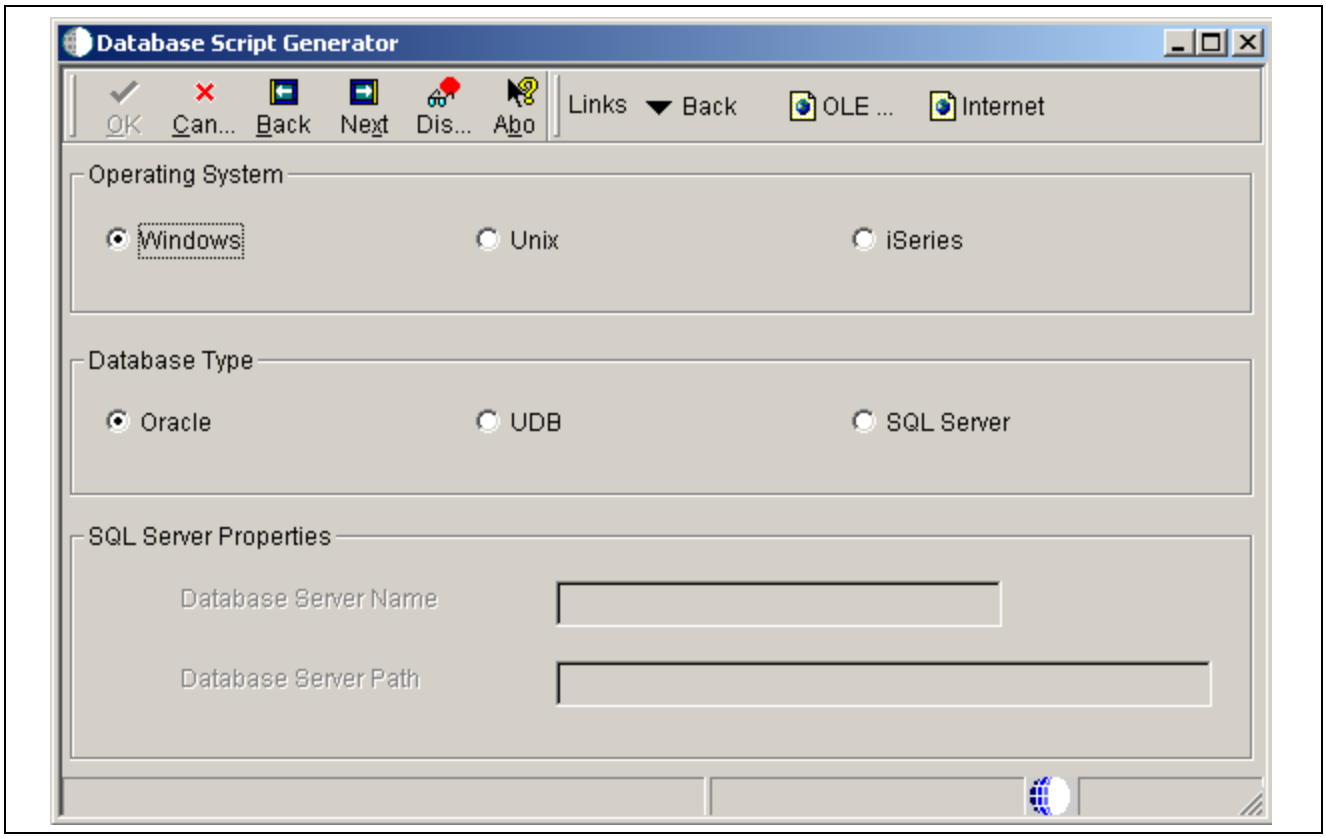
Machine Key Indicate the name of the machine on the network (server or workstation).

Port Number Identify the port for a given instance of JD Edwards EnterpriseOne. Because the JDE.ini file controls the port to which a workstation will connect, for workstations this port number is for reference only.

Data Script Generator

The Data Script Generator form is used to choose your platform and database type.

Access the Data Script Generator form.



Data Script Generator form

Operating System Select the operating system.

Database Type Select the database type.

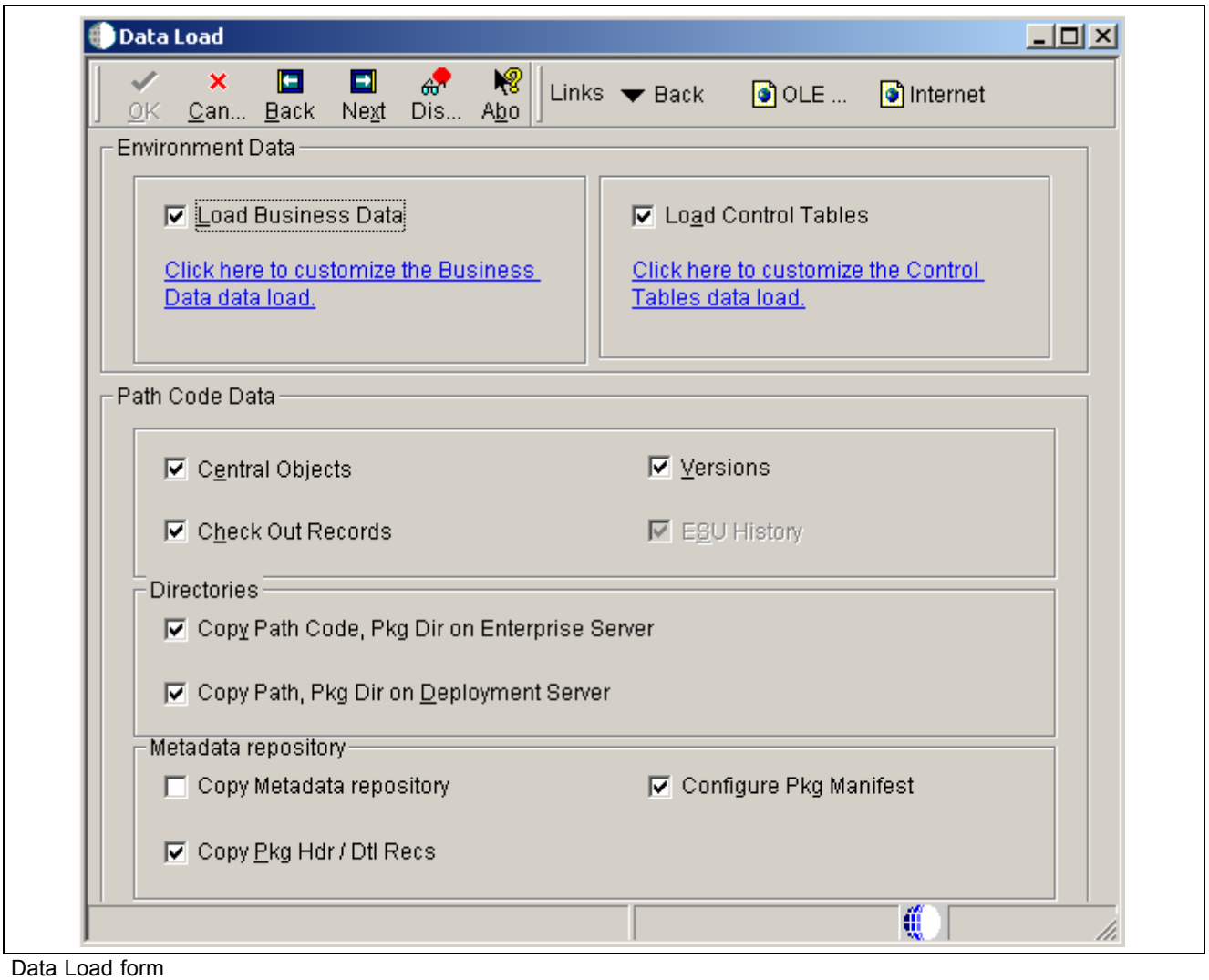
SQL Server Properties Select the SQL server properties.

Data Load

The Data Load form is used to select the data loads the system should perform.

You can override the defaults for copying data. The options for Metadata repository will allow you to copy the source package to the target Central Objects. The process will then configure it as the target package name (rename the repository tables).

Access the Data Load form.



Data Load form

Load Business Data (R98403 XJDE0021) When selected, a form prompts for the source environment.

Load Control Tables (R98403 XJDE0022) When selected, a form prompts for the source environment.

Central Objects and Versions (R98403 XJDE0019) This UBE copies central objects and versions from the source environment to the target environment.

Deployment Server Directory (R9800942 XJDE0001) This copies either the path code on the local file system or the path code on the deployment server. If run from the deployment server, these are the same.

Packages This copies package ini files on the deployment server.

This copies the Software Package Detail (F9631), Software Package Build Header (F96021), and Software Package Build Header - History (F96215) tables.

Check Out Records (R989861 XJDE0001) This checks out the records for the package.

ESU History (R9672 XJDE0001) This copies the ESU history for the package. The system automatically performs this function.

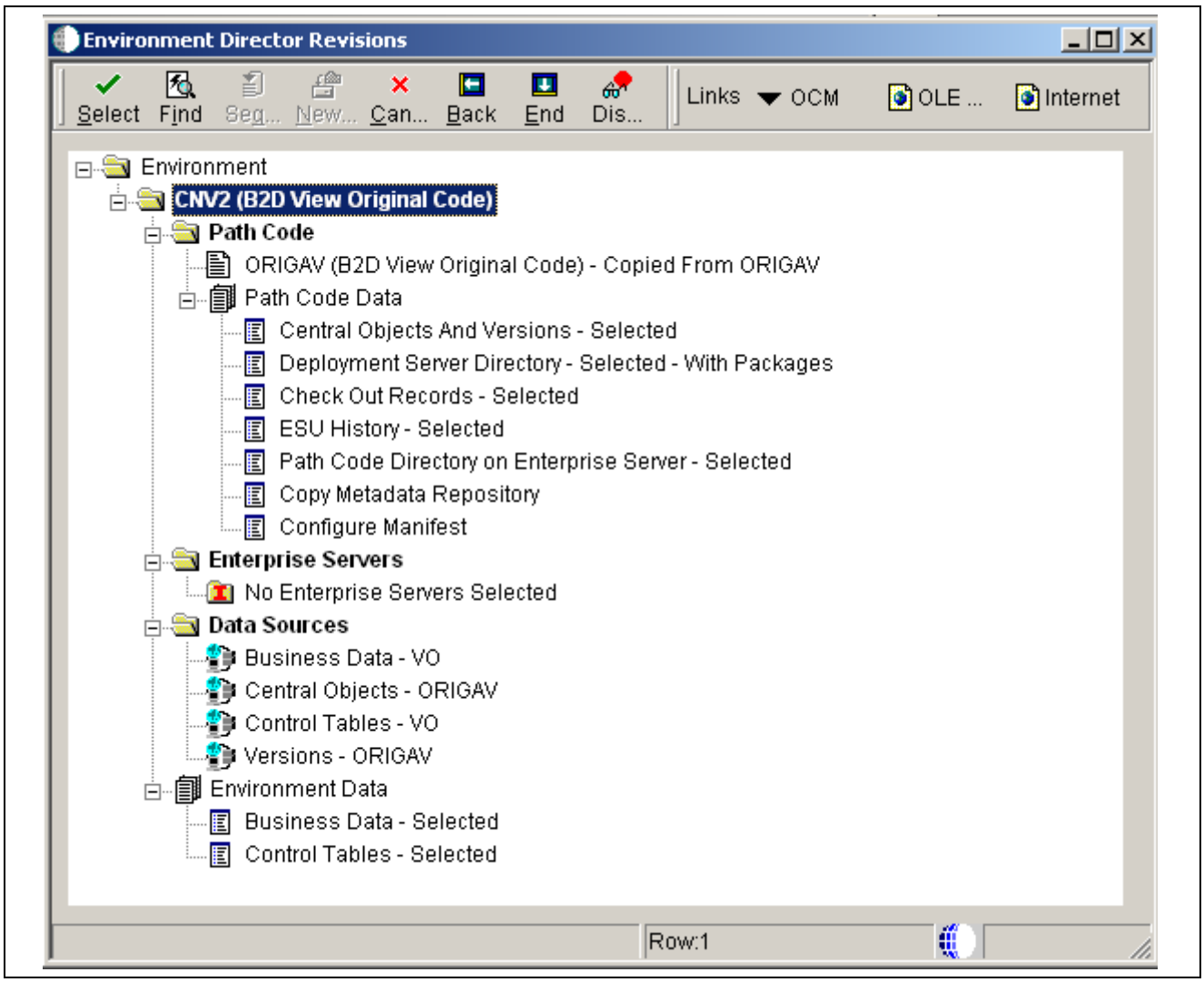
Environment Director Revisions

The Environment Director Revisions form is used to complete the environment creation actions.

At any point in the process you can press the Back button to go back and change an option, but if you click on the cancel button you will lose any answers you have given so far.

The final screen shows you what options were chosen. When you click on End it will start doing the copies. All UBEs are run locally except for the Enterprise Server Path Code copy, which is submitted to the enterprise server.

Access the Environment Director Revisions form.



Environment Director Revisions

Path Code The path code is a pointer to a set of JD Edwards EnterpriseOne objects and is used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Enterprise Servers The names of the computers that receive data from the clients.

Data Sources The names that identifies the data source.

Table Conversion/Merge Log

The application writes history records into F984052. You can see the history by running Table Conversion / Merge Log (application P984052) on menu GH9611. Filter on conversion type 95.

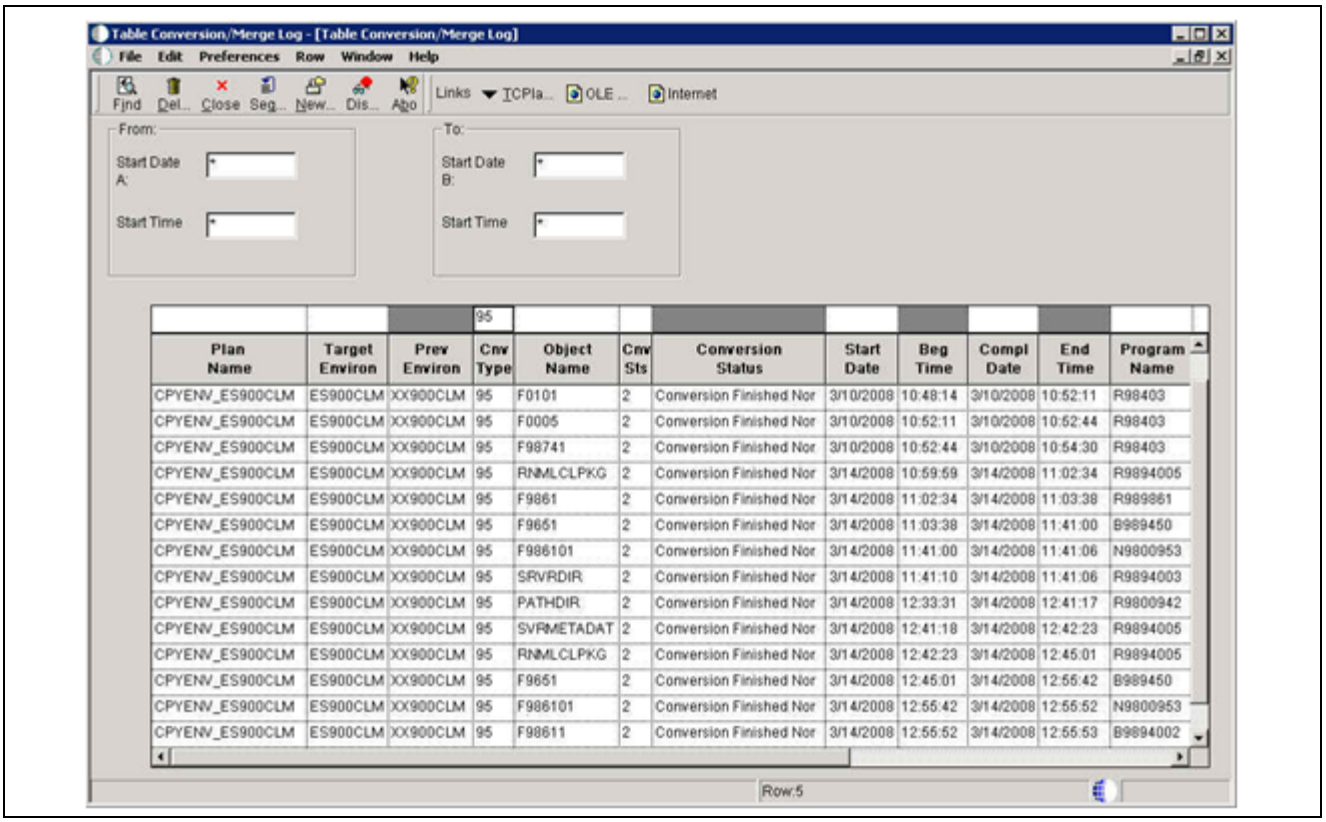


Table Conversion/Merge Log

Using Environment Director in the Express Mode

This section provides an overview of Environment Director in Express Mode and discusses how to create a new environment using the Express Mode.

Understanding Environment Director in Express Mode

You can use the Express mode to create a new environment by using new settings or settings copied from an existing environment. It is recommended that you use director mode rather than express mode to allow for greater control of new environment settings.

The Environment Director Express mode is designed to simplify the process of creating system environments. However, it is recommended that you use director mode rather than express mode to allow for greater control of new environment settings.

Actions Performed

After you enter all the settings for the new environment in Environment Director and click OK, the system performs these actions:

- Creates OCM if it does not already exist.
- Writes F98403 record for the new environment.

Note. This enables Installation Planner (P9840) to display the environment.

- Loads business data.
- Loads control tables.
- Configures enterprise servers.
- Creates server map OCM.
- Creates server map F98611 records.
- Writes F9651 records for enterprise servers.
- Configures BSFN and UBE mappings.

Prerequisites

While Environment Director is a useful tool for creating environments, we recommended using Installation Planner (P9840) and Installation Workbench (P9841) to create custom environments. Environment Director, Installation Planner and Installation Workbench share common functionality for configuring custom environments.

Forms Used with Environment Director in Express Mode

Form Name	FormID	Navigation	Usage
Environment Director	W989400A	In Solution Explorer, navigate to System Installation Tools, Advanced Operations, Environment Director (P989400). Click Next to begin.	Used to step you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.
Environment	W989400B	Complete the fields and click Next.	Used to select the mode to be used for setup of the new environment.
Machines	W989400J	Complete the fields and click Next.	Used to select the enterprise servers where the new environment will run. JD Edwards EnterpriseOne will use this information to populate the server map data sources on these servers.
Environment Director Revisions	W989400L	Verify the fields and click Next.	Used to complete the environment creation actions.

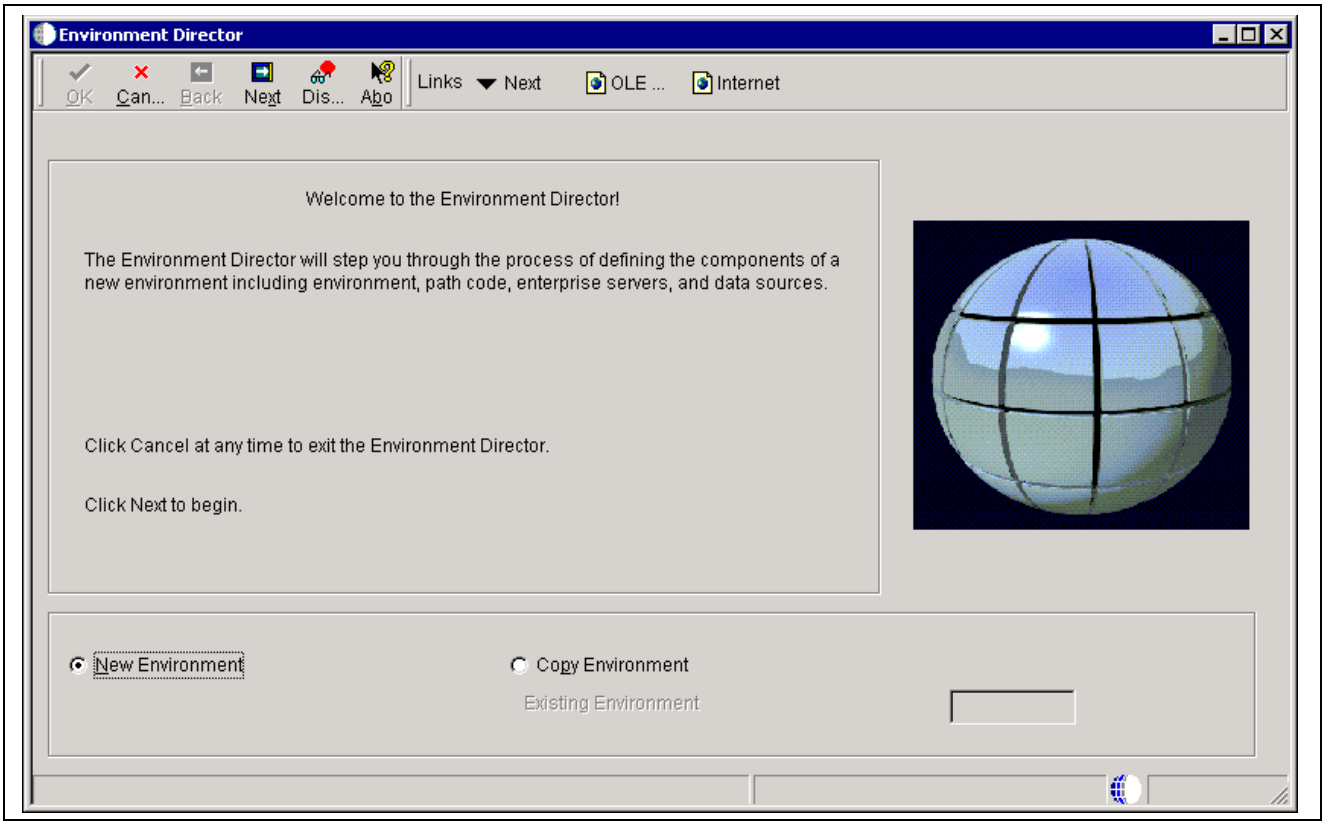
Creating a New Environment in Express Mode

Access the Environment Director form.

Environment Director

The Environment Director steps you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.

Access the Environment Director form.



Environment Director form

New Environment Use this option to create a new environment with no default settings.

Copy Environment Use this option to use settings from an existing environment as the default settings for the new environment. You can modify these settings if needed.

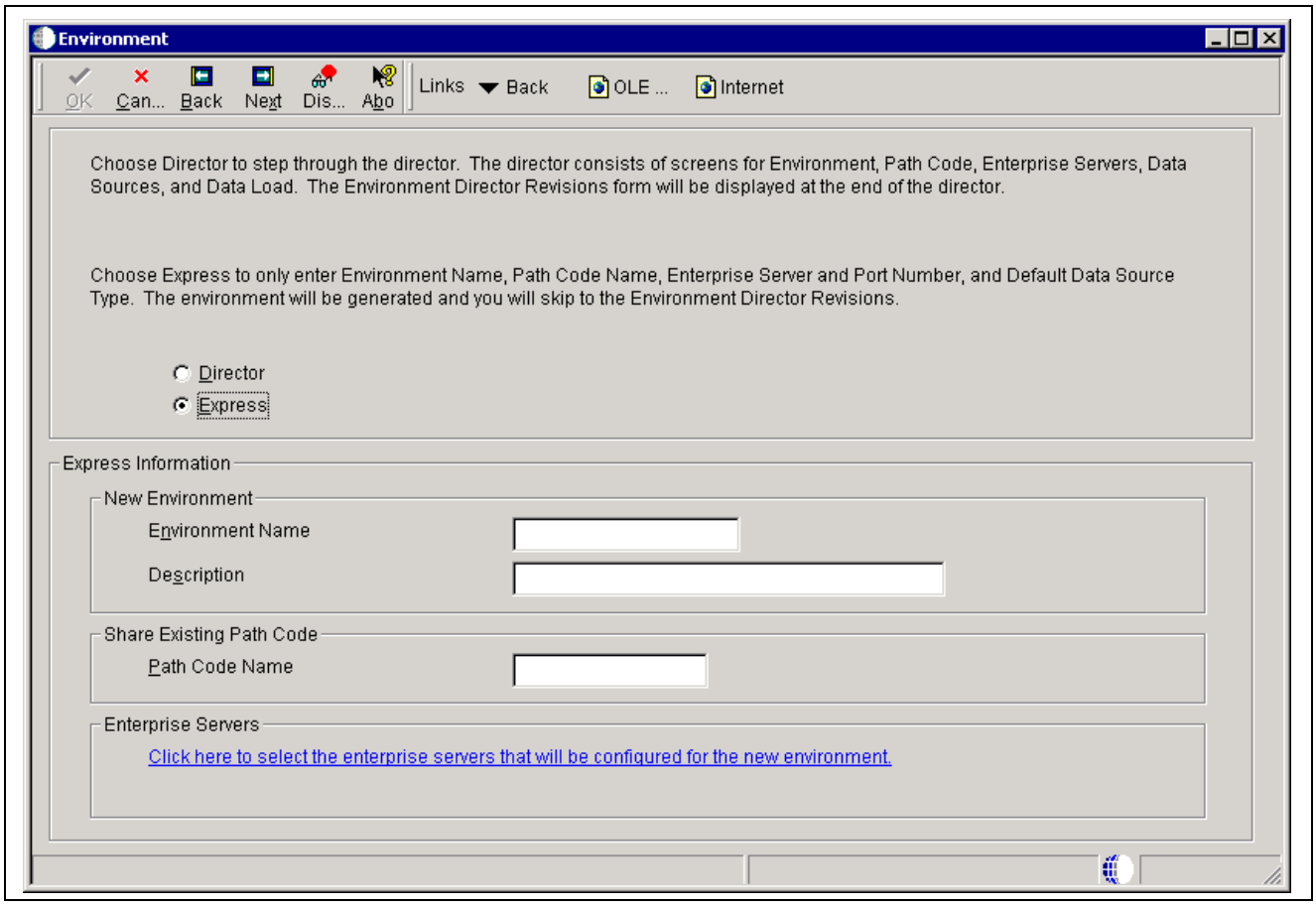
Note. If you selected Copy Environment, click the search button in the Existing Environment field to select an environment from which you want to copy. The existing environment must be at the current release level as the new environment

---

Environment

The Environment form is used to select the mode to be used for setup of the new environment.

Access the Environment form.



Environment form

Director The director mode consists of screens for Environment, path Code, Enterprise Servers, Data Sources, and Data Load. The Environment Director Revisions form will be displayed at the end of the director.

Express The express mode only enter Environment Name, Path Code Name, Enterprise Server and Port Number, and Default Data Source Type. The environment will be generated and you will skip to the Environment Director Revisions.

- New Environment

  Environment Name Enter a valid environment that encompasses both a path code (objects) and a data source (data).

  Description Enter a user defined name or remark.

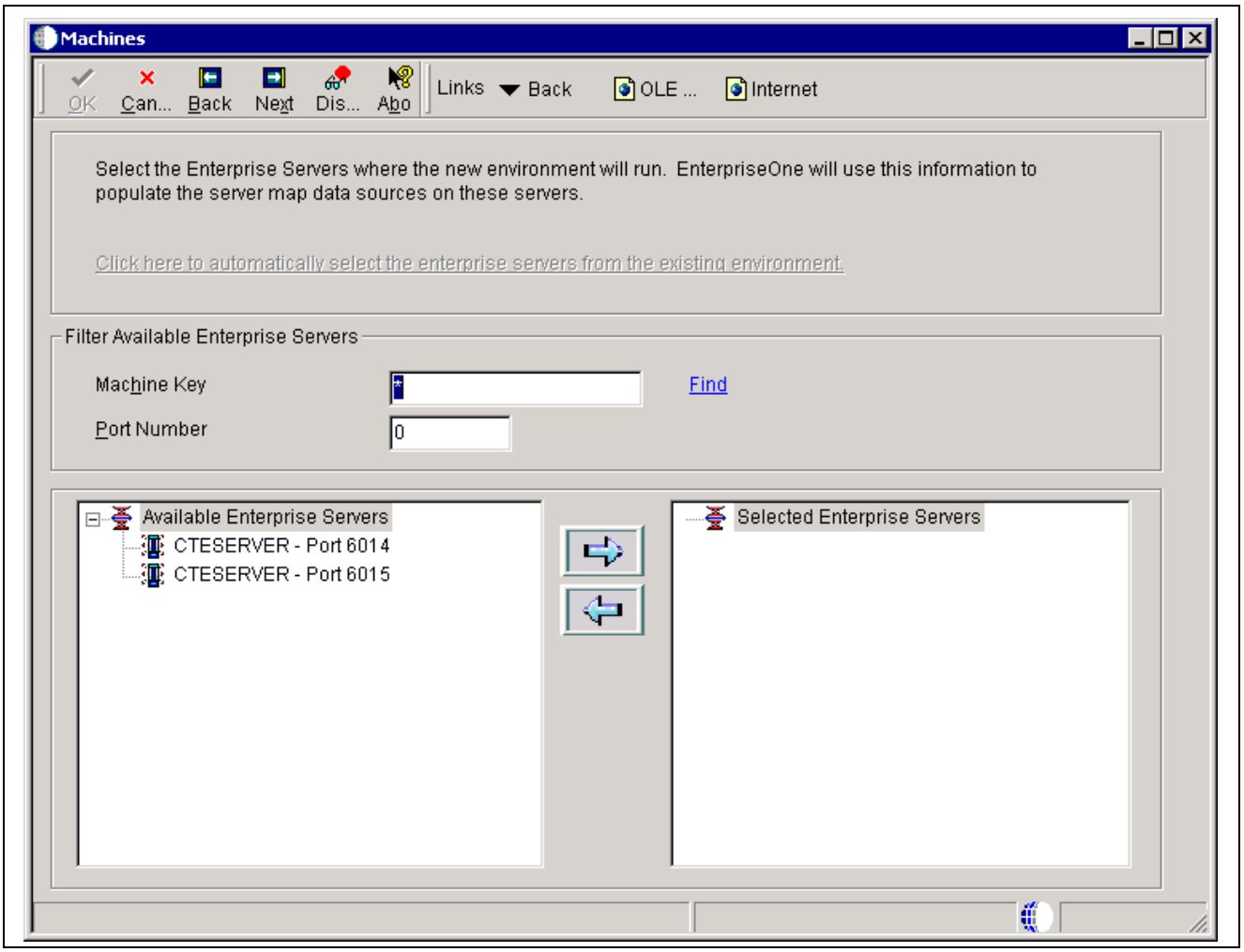
- Share Existing Path Code

  Path Code Name Enter the name of the existing path code.

Machines

The Machines form is used to select the enterprise servers where the new environment will run. JD Edwards EnterpriseOne will use this information to populate the server map data sources on these servers.

Access the Machines form.



Machines form

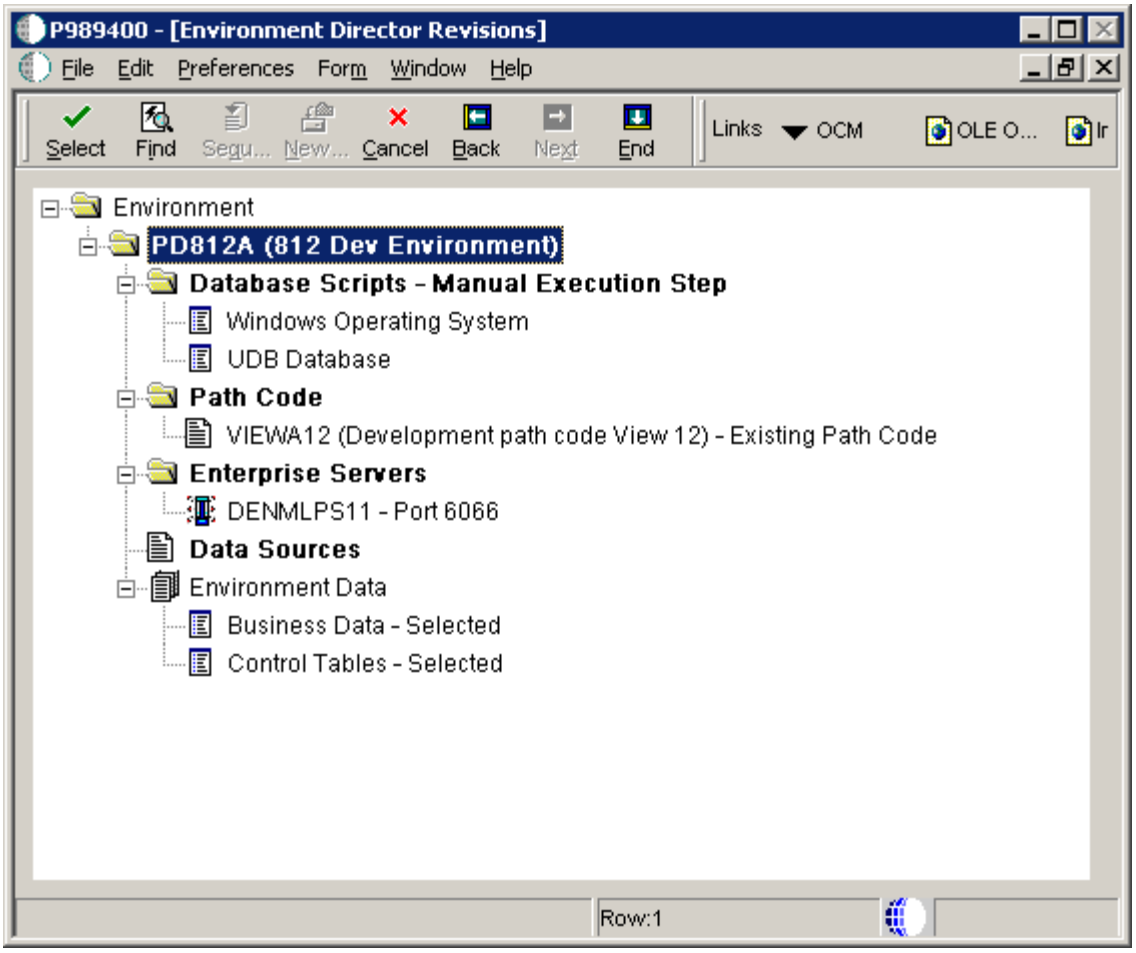
Machine Key Indicate the name of the machine on the network (server or workstation).

Port Number Identify the port for a given instance of JD Edwards EnterpriseOne. Because the JDE.ini file controls the port to which a workstation will connect, for workstations this port number is for reference only.

Environment Director Revisions

The Environment Director Revisions form is used to complete the environment creation actions.

Access the Environment Director Revisions form.



Environment Director Revisions

Path Code The path code is a pointer to a set of JD Edwards EnterpriseOne objects and is used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Enterprise Servers The names of the computers that receive data from the clients.

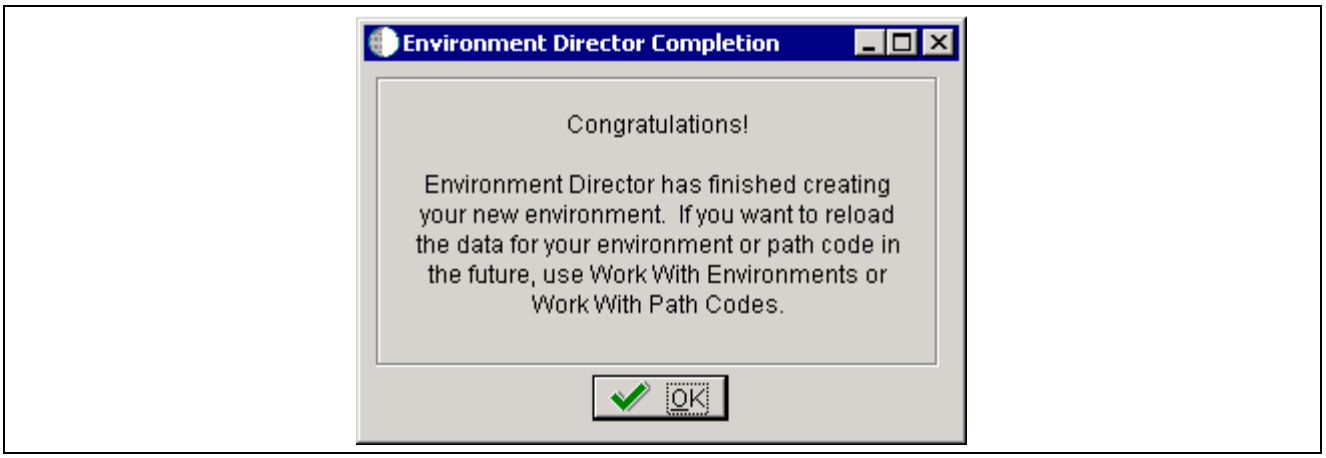
Data Sources The names that identifies the data source.

Environment Director Completion

The Finish Environment Creation message is displayed to let you know that the environment is now ready to be configured. Use Back at any time to verify that the environment properties have been configured correctly before continuing.

Finish Environment Creation message

The Environment Director Completion message is displayed when Environment Director has finished creating the new environment.



Environment Director Completion message

After you enter all the settings for the new environment in Environment Director and click OK, the system will create the new environment.

See the Actions Performed within the Understanding Environment Director in Express Mode section previously in this chapter.

Running Environment Director from Different Environments

This section provides an overview of Running Environment Director from different environments and discusses how to:

- Use Object Management Workbench (P98220) to modify table data classes.
- Use On Track Planning Setup (P98503) to modify table data classes.

Understanding How to Run Environment Director from Different Environments

It is recommended that you create custom environments from the PSFTPLAN environment on the deployment server using Environment Director. Doing so creates the environment in the planner databases and deploys it to the system data source. The environment must exist in the planner databases to apply software updates to the environment.

You cannot run Environment Director from the deployment environment. The deployment environment exists only on the deployment server. It uses the bootstrap tables (F98611 and F986101) from the Planner; the remainder of the system tables are in the system data source. This split will cause inconsistencies between the planner and system definitions.

If you run Environment Director as a standalone, you should run it from a Windows client. The environment will be created in the system data source and needs to be copied to the planner environment before software updates can be applied to the environment. The Copy System to Planner program (R9698611) automates the process of copying the environment definition from system to planner.

Shared Versus Environment Data Source

Environment Director does not prompt for shared data sources. These data sources are shared by all environments and have already been configured and do not need to be reconfigured. By default, these are the System, Object Librarian, and Data Dictionary data sources. Only the environment data sources are configured. By default, these are the Business Data, Control Tables, Central Objects, and Versions data sources.

Database Configuration and Permissions

New databases, table spaces, and libraries must be created before Environment Director can load data into the new environment.

Typically, System, Object Librarian, and Data Dictionary data sources are shared with existing environments and do not need new databases. Business Data and Control Tables data sources, as well as any custom data sources, need new databases if these data sources are not being shared with another environment. Central Objects and Versions need new databases if the new environment will be using a new path code.

See Installing the System Databases in the *JD Edwards EnterpriseOne Applications Release 9.0 Installation Guide (for your platform and database)*.

Understanding Data Source Configuration

This section provides descriptions of various data classes and source templates by environment.

Data Class

The data class is at the center of automatic OCM generation. It categorizes the type of data stored in data sources and tables. For example, Business Data is data class ’B’ and System is data class ’S’. If you want to split business data into multiple data sources, you need to create a new data class, modify the data source template (so Environment Director or Installation Planner prompts for a data source) and assign tables and GT objects to that data class. OCM will be created based on the information provided previously.

Create a New Data Class

You can create a new data class by adding a value to the H96/DU UDC for data sources and H96/CL UDC for table and GT objects.

Data Source Templates by Environment

A data source template defines the set of data sources that will be used by new environments, and the default values for data source properties. The values are stored in the Data Sources by Environment table (F98511).

F98511 also stores the set of data sources for an environment after they have been generated from the template.

The Installation Planner and Environment Director programs use F98511 when prompting for data sources and when creating OCM mappings. When prompting for a data source, the programs will first check the Data Source Master table (F98611) to see whether it has already been defined. If so, Installation Planner and Environment Director will ignore the default settings in F98511 and prompt based on the existing data source definition.

You can edit the information in F98511 using the Data Source Templates By Environment program (P98503). This program enables you to implement custom environment naming standards and ease the creation of environments.

When creating a new environment, the system copies template records to create data sources for the environment and replaces tokens in the properties with values for the environment. Records with an environment value of *SHARED are used as a template for shared data sources. This is used once to set up the shared data sources, whereas new records with environment DEFAULT are created. Environment Director does not prompt for shared data sources; they must be set up from Installation Planner. The default set of shared data sources is system, object librarian, and data dictionary.

Records with an environment *ENV are used for new environment data sources. Whenever a new environment is created, the *ENV records are copied. The environment name replaces *ENV and tokens are replaced in order to receive the actual data source name and default properties. The default set of environment data sources is business data, control tables, central objects, and version. (The central objects data source is also specified on Path Code Master.)

Records with an environment *LOGIC are used for logic and server map data sources. These are set up when selecting an enterprise server.

Table Data Classes

The system uses a field in the Object Librarian Master table (F9860) to associate a table with a data class. You can modify table data classes using either of these programs:

- Object Management Workbench (P98220)

- On Track Planning Setup (P98503)

  Use this program to modify the data class for several tables at once.

Note. Modifying the data class does not affect existing OCM mappings; the data class is only used when creating mappings for a new environment.

Data Load

Environment Director can automate data load for an environment using UBEs. Cases might exist, however, when you want to use third-party tools to load the data (that is, DTS, BCP, Import/Export, CPYLIB). In these cases, the default data load can be de-selected in the director.

Using Object Management Workbench to Modify Table Data Classes

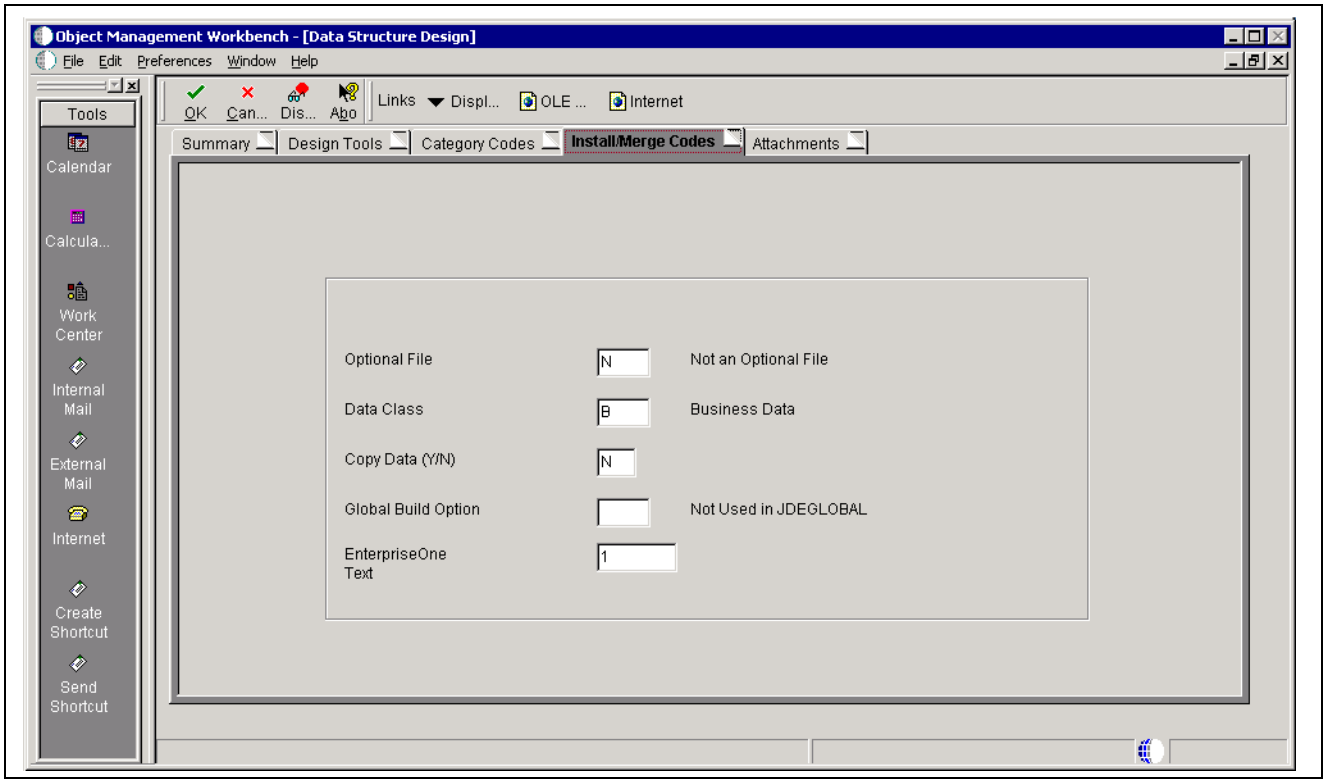
This section discusses how to use Object Management Workbench (P98220) to modify table data classes.

Forms Used to Modify Table Data Classes

Form Name	FormID	Navigation	Usage
Object Management Workbench	W98220A	Application Development, Object Management, Object Management Workbench(P98220).	Used to modify table data classes.
Data Structure Design	W9860AL	In the Object Management Workbench form, click Find, then select an object from the tree in the left pane and click the Design.	Used to modify table data structure.

Using OMW to Modify Table Data Classes

Access the Data Structure Design form.



Data Structure Design form

Optional File Specify the name of the optional file. Values are:

Y - Designates a file as an Optional Data File if there are some situations where the file may not be needed at a client installation. The explanation of these situations can be found in the Generic Rate/Message information for that file for Generic Rate/message Type 96/OF. All of these files that exist in a specified library can be listed in the Optional File Report on menu A9645.

O - Designates that the file is designated for omission. Examples are compile files or special files like JDE User Profiles file.

N - Not an Optional File.

Data Class Select a code that classifies the type of data that is stored in a JD Edwards EnterpriseOne table. The standard classes are System, data Dictionary, Object Librarian, Business data, Control Tables, Central Objects, and Versions. When you create or modify OCM mappings for an environment, this value is used to associate this table to a JD Edwards EnterpriseOne data source (DATUSE).

Values are:

B - Business Data

C - Central Objects

D - Data Dictionary

I - Internal

O - Object Librarian

P - Installation Planner

S - System

T - Control Tables

V - Versions

Copy Data (Y/N) Indicate if a file and its data are copied into production. A value of N moves the file without data during a file copy. When creating a production data library from JDFDATA, this field designates whether the data is included in the copy.

Global Build Option Future Use.

JD Edwards EnterpriseOne Text Future Use.

Using On Track Planning Setup to Modify Table Data Classes

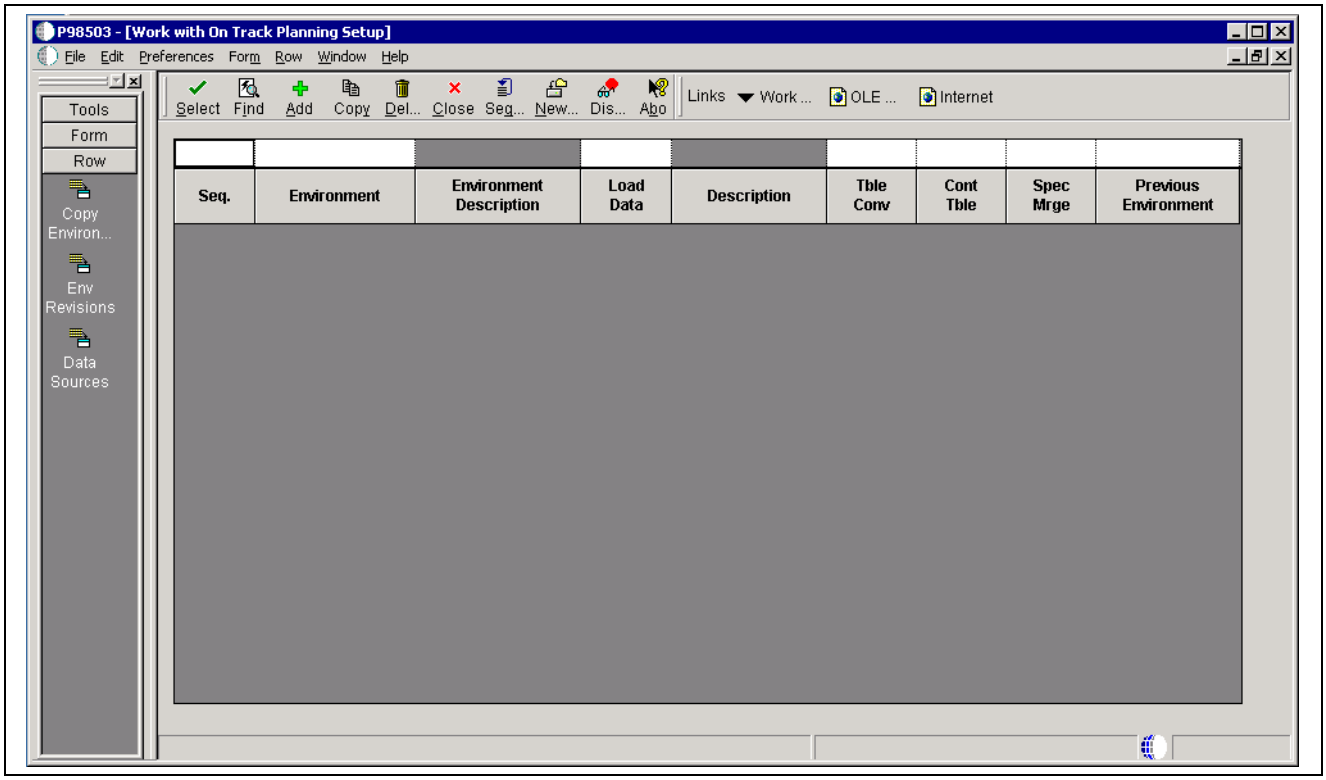
This section discusses how to use On Track Planning Setup (P985033) to modify table data classes.

Form Used to Modify Table Data Classes

Form Name	FormID	Navigation	Usage
Work With On Track Planning Setup	W98503A	In Solution Explorer, navigate to System Administration Tools, System Installation Tools, Advanced Operations, On Track Planning Setup(P98503).	Used to modify table data classes.

Using On Track Planning Setup to Modify Table Data Classes

Access Work With On Track Planning Setup form.



Work With On Track Planning Setup form

Seq. Future use.

Environment Specify the name associated with a specific list of libraries. The J98INITA initial program uses these library names to control environments that the user can sign on to. These configurations of libraries lists are maintained in the Library List master File table (F0094).

This field represents a valid environment that can be used to run in JD Edwards EnterpriseOne. The environment encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace within the system.

Environment Description Specify a user defined name or remark.

Load Data	Denote the type of data that is loaded for an environment. Values are: *0* - No Data is loaded. *1* - Production data is loaded. Tables that are marked in the Object Librarian to copy data, such as constants tables, will load data in the table: the other data/transaction tables are created empty. *2* - Demonstration data is loaded. All tables are copied in.
Description	Specify a user defined name or remark.
Tble Conv	Denote whether the table conversions are performed for this environment.
Cont Tble	Denote whether the table merges are performed for this environment.
Spec Mrge	Denote whether the specification merges are performed for this environment.
Previous Environment	Specify the previous Environment is the existing environment that will be used as a base to create the new environment.

CHAPTER 9

Understanding Object Configuration Manager

This chapter provides an overview of Oracle's JD Edwards Object Configuration Manager and discusses:

- Distributed architecture
- Partitioning logic on servers

Object Configuration Manager

This section provides and overview of Object Configuration Manager functionality and discusses:

- OCM Characteristics
- OCM Information Requests
- Object Mappings

OCM Functionality

The OCM provides the flexibility to map data, batch applications, and business functions to a data source, which enables you to coordinate the distributed processing. For example, you would map table objects to database data sources and logic objects to machine data sources

This table describes the minimum of two sets of the Object Configuration Master (F986101) and Data Source Master (F98611) tables that you must have:

One for All Workstations	The Object Configuration Master and Data Source Master tables that the software uses for workstation processing are stored in the centralized system data source normally kept on an enterprise server. If the system data source is not available, the software looks to the workstation's jde.ini file for a secondary location.
One per Logic Server	The Object Configuration Master and Data Source Master tables that the logic server uses are stored on that server in the server map data source. Each logic server type requires its own server map data source.

OCM Characteristics

OCM contains a large number of mappings that define where individual tables exist when looking for data. OCM also contains mappings that define where particular types of logic, specifically business functions or UBEs, will be processed. When a request for data or logic processing occurs, OCM directs it to the appropriate database or machine.

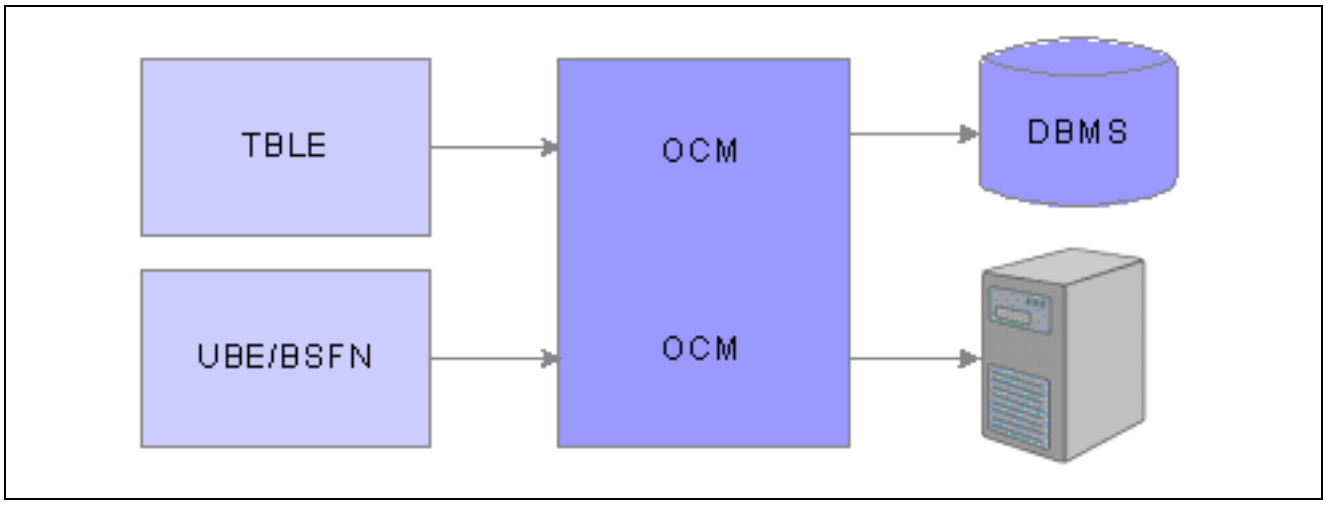
OCM has these characteristics:

- Data and logic resources can be located on any machine, and JD Edwards EnterpriseOne must know where each resource is located.
- OCM is the method of managing distributed data and distributed logic processing in a CNC configuration
- OCM is a collection of data sources, or pointers, to databases and logic processing machines

OCM Information Requests

When an application requests data from a table or a logic object for processing, OCM points to the appropriate database data source.

This diagram illustrates an information request from OCM.



OCM Information Requests

Object Mappings

In Oracle's JD Edwards EnterpriseOne, business objects are used to configure distributed processing and distributed data at runtime. The Object Configuration Master table defines this configuration. You can work with the server object map to modify the entries in this table.

Workstation mappings are stored in a centralized system data source. The F986101 table used by the enterprise server is stored on that server in the server map data source. Each enterprise server requires a separate server map data source.

Compared to a workstation, an enterprise server processing a logic object has a different perspective of where data should be retrieved. For a workstation that is requesting user defined codes, its F986101 table (in the system data source) points to the local database.

When an enterprise server requests user defined codes, it makes no sense for the enterprise server to look to a workstation for this information; therefore, the enterprise server should have unique mappings for user defined codes. These mappings are set up in the Object Configuration Master table (F986101) in the server map data source.

If you have changed table F986101 for the workstation, you should check it in the server maps to see if they should also be changed. For example, if you have new environments with unique mappings for the workstation, you should check to see if changes are required in the corresponding mappings for the enterprise server.

The Object Configuration Manager also provides batch processes to help with the administration of the object mappings. These processes perform such tasks as comparing, updating, copying, and deleting Object Configuration Manager records.

Mapping Alternatives

You map objects by environment. You select an environment that you have already created and map that environment's objects to the data sources you want those objects to use. You can set default mappings for all instances of an object type to one data source, and you can map individual objects to data sources.

This table describes mapping alternatives:

Mapping Object Types: Default Maps	To create a default map for an object type, create a mapping whose object name is the literal value: DEFAULT. Then enter an object type (such as TBLE) and a data source. By creating a default map for the object type TBLE, any table objects not mapped individually point to the default data source.
Mapping Individual Objects	You can map individual objects within an environment. For example, you can map a specific table, such as the Security Workbench Table (F00950), to a data source other than the default, such as to the system data source. If you do not explicitly map an object by name in the Object Configuration Manager, the software uses the default map for that object's type.

Important! Each environment must have a default map for TBLE (table) objects for the *PUBLIC user profile because there is no inherent default location for table objects. If table objects do not have a default map and are not explicitly mapped by name, the software produces a Select/Failed error message.

Distributed Architecture

The software enables you to distribute data and logic in a manner that optimizes both the power of the workstation and the data integrity of the server. This optimization provides you with:

Flexibility in determining your own partitioning schemes.	As requirements change, you can repartition the system quickly and easily to meet new needs.
Independence in using the data and logic objects that you need.	For example, if you are a salesperson on the road, you can download only the data and logic you need to quote prices and availability. Later, you can connect to the server and place the orders in a batch process.
Growth for your enterprise systems.	The system can be as large as you need because the software enables you to move objects around the system in practically endless combinations.

To keep track of where data resides and where logic processing occurs, the software uses a tool called the Object Configuration Manager. This tool enables users to specify data and logic processing locations.

Partitioning Application Logic on Servers

The logic for JD Edwards EnterpriseOne applications can be partitioned to run remotely by mapping individual or specified groups of business function components to run on an application server or enterprise server instead of on a workstation.

It has been found that redeploying certain business function components (including master business functions and business functions) can significantly increase the performance of a distributed JD Edwards EnterpriseOne workstation while simultaneously decreasing network traffic. This redeployment involves remapping objects using the JD Edwards EnterpriseOne standard Object Configuration Manager methodology.

Examples of such configurations are illustrated by the Windows light client/heavy server and the Java light client/heavy server models. Both models have applications specifications on the client and business function components on the server. The main difference is that the Windows model uses JDENet communication middleware, while the Java model uses JDENet/CORBA middleware.

While the software design enables you to partition all business function components, the biggest benefit is derived from partitioning Master Business Functions (MBFs).

JD Edwards EnterpriseOne transaction-oriented applications are built around the concept of MBFs, which are typically responsible for transaction edits and for committing transactions to the database. Most of the I/O services for transaction-oriented applications are performed by MBFs. By localizing the majority of business logic for transactions in MBFs and partitioning the MBFs to run on application servers, network traffic can be minimized, thus dramatically improving the performance of the application in distributed and WAN environments.

In a two-tier setup where MBFs are processed on the client, a lot of interaction occurs across the WAN between client and server. In a three-tier setup consisting of a client, a data server, and an application or enterprise server, transaction processing can occur across a LAN between the two servers. Interaction across the WAN between the client and server is thus reduced to entering input on the client and sending back results from the server. This three-tier configuration can result in a significant reduction in traffic across the WAN.

The following contrasts typical network traffic for a two-tier setup where MBFs are processed on the client versus network traffic segmentation for a three-tier setup where MBFs are processed on the server.

Two-Tier: Typical Network Traffic

In a two-tier configuration, the GUI, event rules, and MBFs are typically handled by the client, and data is stored on the server. Typically, this processing occurs across the WAN between client and server:

- Fetch Record (client to server)
- Return Record (server to client)
- Validate Data Format (client to server)
- Format OK (server to client)
- Send Record Detail (client to server)
- Detail OK (server to client)
- End Transaction (client to server)

Three-Tier: Network Traffic Segmentation

In a three-tier configuration, the GUI and event rules are handled by the client, but an application server or enterprise server handles MBF processing. The database server stores data. This processing occurs across the WAN between client and application or enterprise server:

- Input Processing Request or Data (client to server)
- Return Processing Results (server to client)

This processing occurs locally across the LAN between the application or enterprise server and the database server:

- Fetch Record
- Return Record
- Validate Data Format
- Format OK
- Send Record Detail
- Detail OK
- End Transaction

Master Business Function Operations

This series of events demonstrates how a typical application uses a Master Business Function (MBF). This example uses the Sales Order Entry application.

End of Sales Order Line

The first event occurs when the end of a sales order line is reached, causing the JD Edwards EnterpriseOne client application to call the jdeCallObject API. This command sends a message to the MBF. Included with the message is data (in the form of a data structure) for the line. The application sends the message asynchronously with its associated data; that is, once the message is sent, the client application proceeds to the next line.

MBF Receives Line Message

This event occurs when the MBF receives the JDENet message that includes the data for the line. The line data is cached in the server's shared memory.

MBF Extends and Edits the Line

This event occurs when the MBF extends and edits the sales order line. The data necessary to extend and edit the line is typically accessed locally on a LAN. The data is requested by a database-dependent SQL call and is transported by the applicable Open Database Connectivity (ODBC) or Oracle Call Level Interface (OCI) mechanisms.

MBF Sends a Return Message to the Client Application

This event, the fourth event, occurs after the MBF extends and edits the sales order line and returns the extended line, as well as any error codes, to the client. The return message is sent using JDENet. Events 1 through 4 are then repeated asynchronously for all of the lines associated with the sales order.

End of Sales Order (OK Button)

This event indicates that the user has completed all sales order lines. The user triggers this event by clicking OK after all edited lines have been returned to the client. When the user clicks OK, an end of transaction message is sent to the MBF. The client is immediately released to enter the next transaction.

MBF Processes the Full Transaction

The full transaction is processed when the MBF asynchronously reads the shared memory cache (where all transaction lines are stored) and begins the process of committing the transaction to the database.

Transaction Commitment to the Database and MBF Cleanup

The MBF commits the entire transaction to the database, typically locally through ODBC and OCI, and cleans up the shared memory cache for the completed transaction.

Mapping the MBF to run on the server causes the bulk of the database and logic interaction to occur within a single server machine (enterprise server) or between LAN-attached machines (application server and data server). Thus the transaction has been processed with a minimum of network traffic. This type of application transaction is ideally suited for performance gains in distributed and WAN environments.

CHAPTER 10

Working with Object Configuration Manager

This chapter provides and overview of the Oracle JD Edwards Object Configuration Manager program and discusses how to:

- Work with Object Configuration Manager
- Set up object mappings
- Change mappings for an Object Librarian table
- Update the Object Configuration System table
- Create OCM records for business functions
- Update the Oracle Parameters table.
- Revise the Generic Text Language Status table
- Real Time Events (RTE)
- Workflow Events (WFE)
- Extended Third-Party API (XAPI)

Understanding Object Configuration Manager

The Object Configuration Manager (OCM) program (P986110) has the ability to distribute logic and data provides for flexibility in determining your own partitioning schemes. It also provides independence in using only the data and logic objects that you need and allows for growth within your enterprise systems. You can later add more databases to store data or machines to process logic. You would need to define the data sources for each and create appropriate OCM mappings.

The OCM stores information in tables that tell the software where data resides and where processing occurs. At runtime, the software looks to the OCM to determine these data and processing locations.

OCM configures distributed processing and data dynamically without any programming. Depending on the environment and the user, the OCM points to the correct location for:

- Data
- Batch processes
- Business functions
- Events

In Oracle's JD Edwards EnterpriseOne, business data objects (tables) map to database data sources. Batch processes and business functions map to machine data sources. Events map to datasources.

You always need at least two OCM tables:

One table for all workstations. Store this table in a centralized system data source. Normally, a central data server stores the system data source. If the central server is unavailable, JD Edwards EnterpriseOne looks to the workstation's jde.ini file for a secondary location.

One table for each logic server. Servers process differently than workstations; for example, the server map data source for each logic server stores separate OCM tables for server processing.

Example: Application Request for Logic

This is an example of how the OCM works with a general logic request.

Step 1: The request process for application logic is similar to data requests. The OCM controls where all business functions and batch processes are processed.

For example, when you add a purchase order, any event that calls a business function looks to the OCM to determine where to process that business function. After you click OK to complete a purchase order, the software calls a master business function to validate all information and record the transaction. The software can process these business functions locally or on the server.

The primary, unique index to the OCM includes:

- Environment, such as PD900 or DV900.
- User, which is either a specific User ID / Role or *PUBLIC.
- Object name, such as F0101, B401002, or R09801.
- Database path

For this example:

- The environment is PD900 (production)
- The status is AV (active)
- The object type can be either a batch process (UBE) or business function (BSFN)

This table describes the search hierarchy that the OCM uses to locate the correct data source for a logic request:

Search Sequence	Object Name	User or Role	Search Criteria
1	B0900049	SI5745669 (user ID)	Is there a record for the named environment, status active, type (UBE or BSFN) for the named object, and the specific user?
2	B0900049	OWTOOL (role)	Is there a record for the named environment, status active, type (UBE or BSFN) for the named object, and the specific role?

Search Sequence	Object Name	User or Role	Search Criteria
3	B0900049	*PUBLIC	Is there a record for the named environment, status active, type (UBE or BSFN) for the named object, and *PUBLIC?
4	DEFAULT	SI5745669 (user ID)	Is there a record for the named environment, status active, type (UBE or BSFN) with no named object (default), and the specific user?
5	DEFAULT	OWTOOL (role)	Is there a record for the named environment, status active, type (UBE or BSFN) with no named object (default), and the specific role?
6	DEFAULT	*PUBLIC	Is there a record for the named environment, status active, type (UBE or BSFN) with no named object (default), and *PUBLIC?
7	NA	NA	If there is no record for this object type, then the software processes the process on the workstation.

Step 2: After the data source is determined, the software passes the definition of that data source to JDENet.

Step 3: JDENet sends a message to the server to begin processing logic. When JDENet on the server receives the message, a JD Edwards EnterpriseOne process on the server responds to the message by processing the requested logic object.

Setting Up Object Configuration Manager

This section discusses how to set up processing option for Object Configuration Manager

Form Used to Map Objects

Form Name	FormID	Navigation	Usage
Object Configuration Manager	W986110D	In the Microsoft Windows client, in Solution Explorer, navigate to System Administration Tools, Object Configuration Manager (P986110). In the web client enter P986110 in the Fast Path.	Determine the location where data is located and logic is run.

See Also

JD Edwards EnterpriseOne Tools 8.98 System Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools System Administration"

Setting Processing Options for Object Configuration Manager

The Object Configuration Manager program (P986110) has one processing option that controls error handling.

Process

Although processing options are set up during JD Edwards EnterpriseOne implementation, you can change processing options each time you run a program.

1. Non-existent table error Specify whether an error or a warning should be issued when a table does not exist in the data source to which it is mapped. Values are:

1 Issue an error

Blank: Issue a warning

Setting Up Object Mappings for the Object Librarian Table

This section discusses how to set up object mappings for an Object Librarian table.

Forms Used to Set Up Object Mappings for the Object Librarian Table

Form Name	FormID	Navigation	Usage
Machine Search & Select	W986110D	In Solution Explorer, navigate to System Administration Tools, System Installation Tools, Advanced Operations, Object Configuration Manager (P986110). In the web client enter P986110 in the Fast Path.	Used to display the data source that stores the Object Configuration Manager table.
Work With Object Mappings	W986110B	In the Machine Search & Select form, select the data source you want and click Select.	Used to display data sources that have the OCM Data Source field checked on the Data Source Revisions form.
Object Mapping Revisions	W986110C	In the Work With Object Mappings form, click Add.	Used to add new object mappings for data sources.

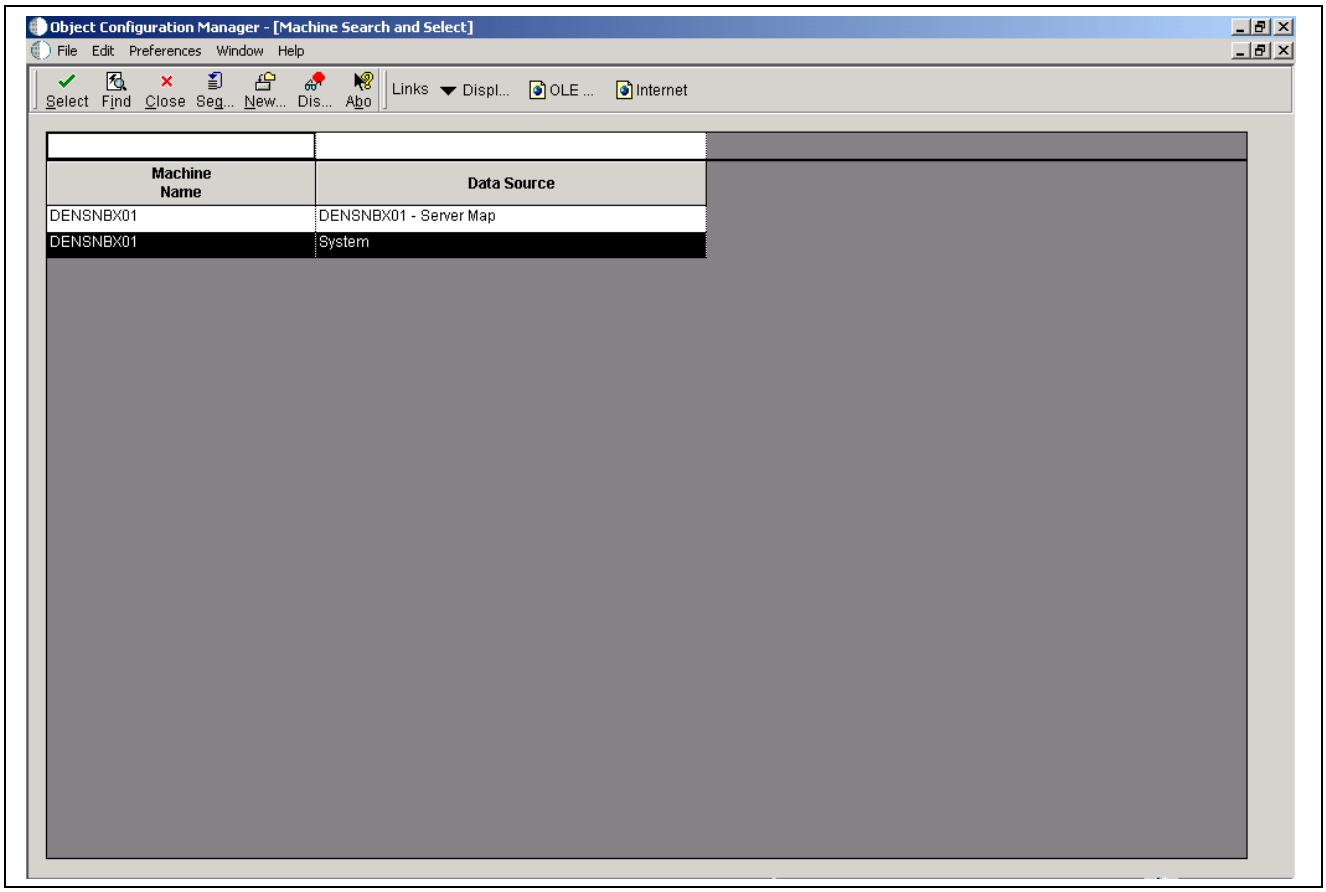
Setting Up Object Mappings for the Object Librarian Table

Batch applications and business functions automatically run locally if there is no default map for that object type.

Machine Search & Select

The Machine Search & Select form displays data sources that have the OCM Data Source field checked on the Data Source Revisions form. Workstation use the system data source for their mappings. Each enterprise server has its own server map data source.

Access the Machine Search & Select form.



Machine Search & Select form

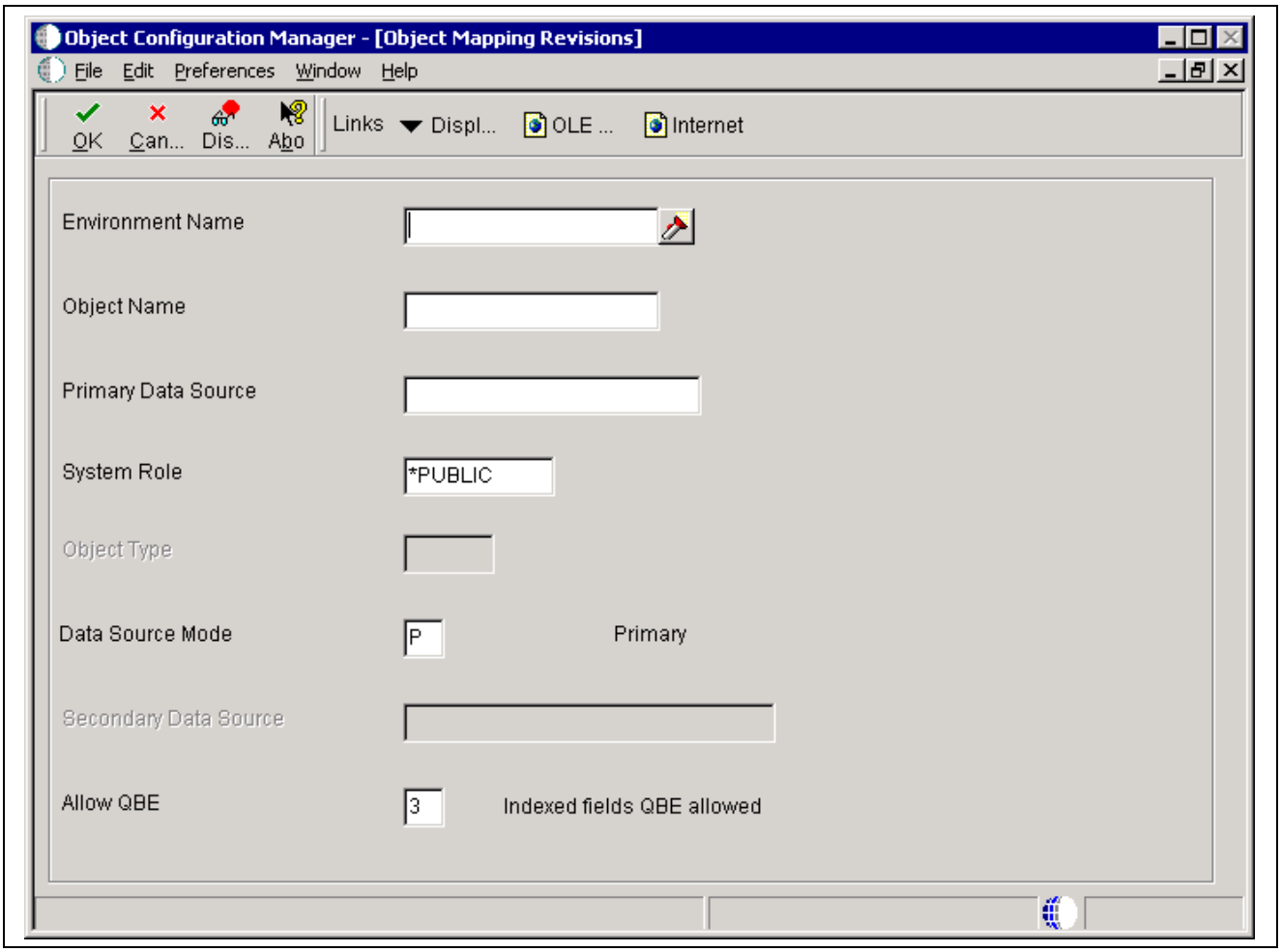
Machine Name A value that defines the logic host to JD Edwards EnterpriseOne.

Data Source The name that identifies the data source.

Object Mapping Revisions

Object Mapping Revisions is used to add new object mappings for data sources.

Access the Object Mappings Revisions form.



Object Mapping Revisions form

Environment

Specify the environment name is also called the Plan Name and is used to uniquely identify an upgrade environment for Install/Reinstall.

For JD Edwards EnterpriseOne (Environment or Version Applications) this is the path code that identifies the location of the application or version specification data.

Object Name

Specify the name that identifies a system object. JD Edwards EnterpriseOne ERP architecture is object-based. Discrete software objects are the building blocks for all applications. The Object Librarian tracks each object. Examples of system objects include:

- Batch Applications
- Interactive Applications
- Business Functions
- Business Functions Data Structures
- Event Rules
- Media Object Data Structures

Object Type Specify the type of object with which you are working. For example, if you are working with tables the object type is TBLE, or business functions is BSFN.

BSFN Business Function Modules

GT Generic Text / Media Objects

RTE Real Time Event

TBLE Table Definition

UBE Batch Applications

XAPI Extended Third-Party API Call

Primary Data Source Specify the name that identifies the data source.

System Role Specify a profile that you use to classify user into groups for system purposes. You use group profiles to give the members of a group access to specific programs.

Some rules for creating a profile for a user class or group are as follows:

- The name of the user class or group must begin with and Asterisk (*) so that it does not conflict with any system profiles.
- The User Class/Group field must be blank when you enter a new group profile.

Object Status Indicate the data source called by the specifies object.

Secondary Data Source Use this data source if the primary data source or the data item in the primary data source cannot be located.

Allow OBE Use this flag to turn On or OFF row-level record locking for the data source.

You should have this flag turned ON to help prevent database integrity issues.

JDEBASE middleware uses this flag to determine whether or not to use row-level record locking.

Changing Mappings for an Object Librarian Table

This section discusses how to change object mappings for an Object Librarian table.

Forms Used to Change Object Mappings

Form Name	FormID	Navigation	Usage
Machine Search & Select	W986115E	In Solution Explorer, navigate to System Administration Tools, System Installation Tools, Advanced Operations, Object Configuration Manager (P986110).	Used to display the data source that stores the Object Configuration Manager table.
Work With Object Mappings	W986110B	In Machine Search & Select form, select the data source and click Select.	Used to display data sources that have the OCM Data Source field checked on the Data Source Revisions form.
Revise OL Data Source	W986110G	In the Work With Object Mappings form, click Revise OL DS from the Form menu.	Used to map Object Librarian tables to a data source.

Changing Mappings for an Object Librarian Table

When you map any of the Object Librarian tables, the software validates your entries to ensure that all environments based on the same path code have their Object Librarian tables mapped to the same data source. The software alerts you with an error message if you map your Object Librarian tables to different data sources.

Work With Object Mappings

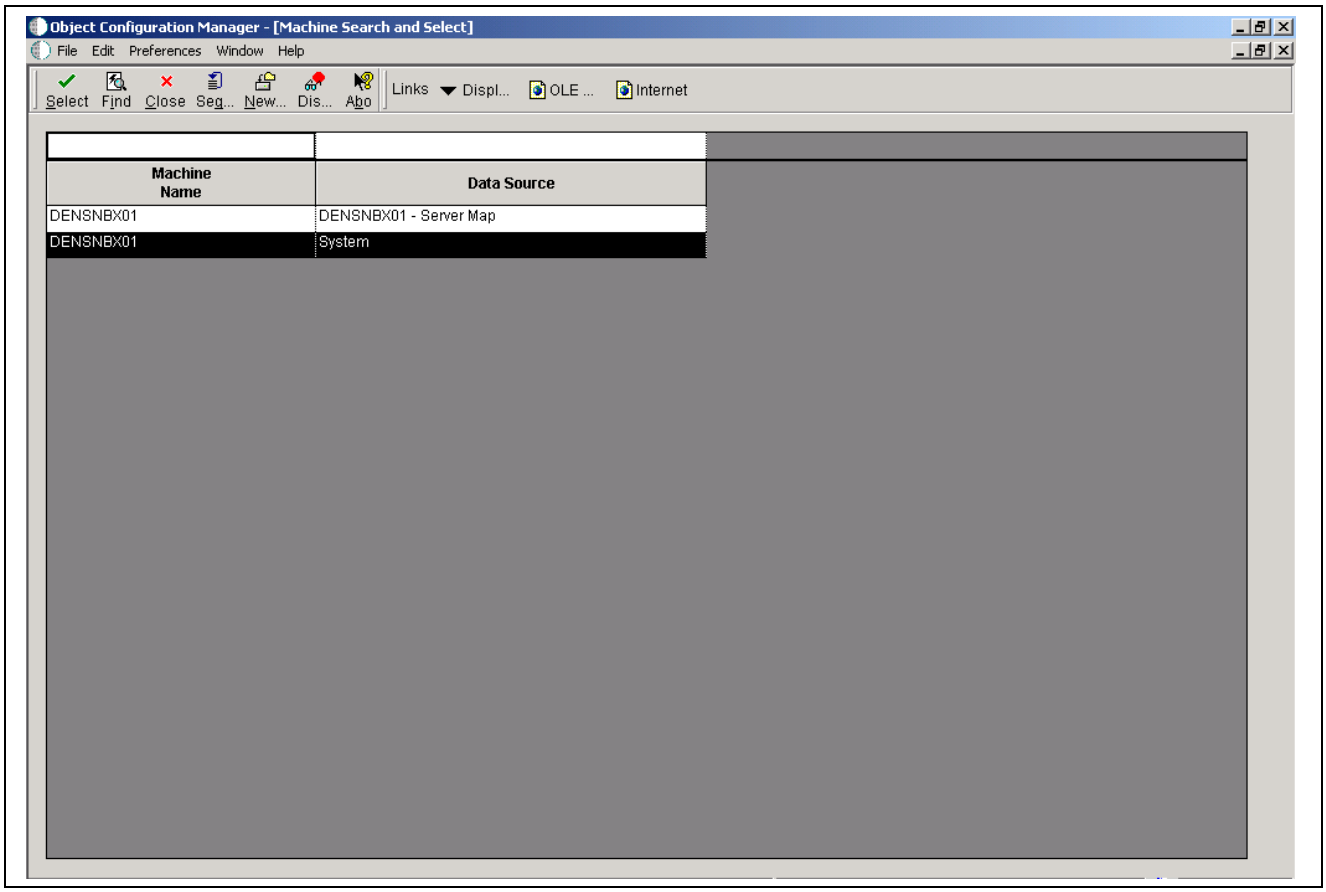
The Work With Object Mappings form is used to display data sources that have the OCM Data Source field checked on the Data Source Revisions form.

Access the Work With Object Mappings form.

Machine Search & Select

The Machine Search & Select form displays data sources that have the OCM Data Source field checked on the Data Source Revisions form. Workstation use the system data source for their mappings. Each enterprise server has its own server map data source.

Access the Machine Search & Select form.



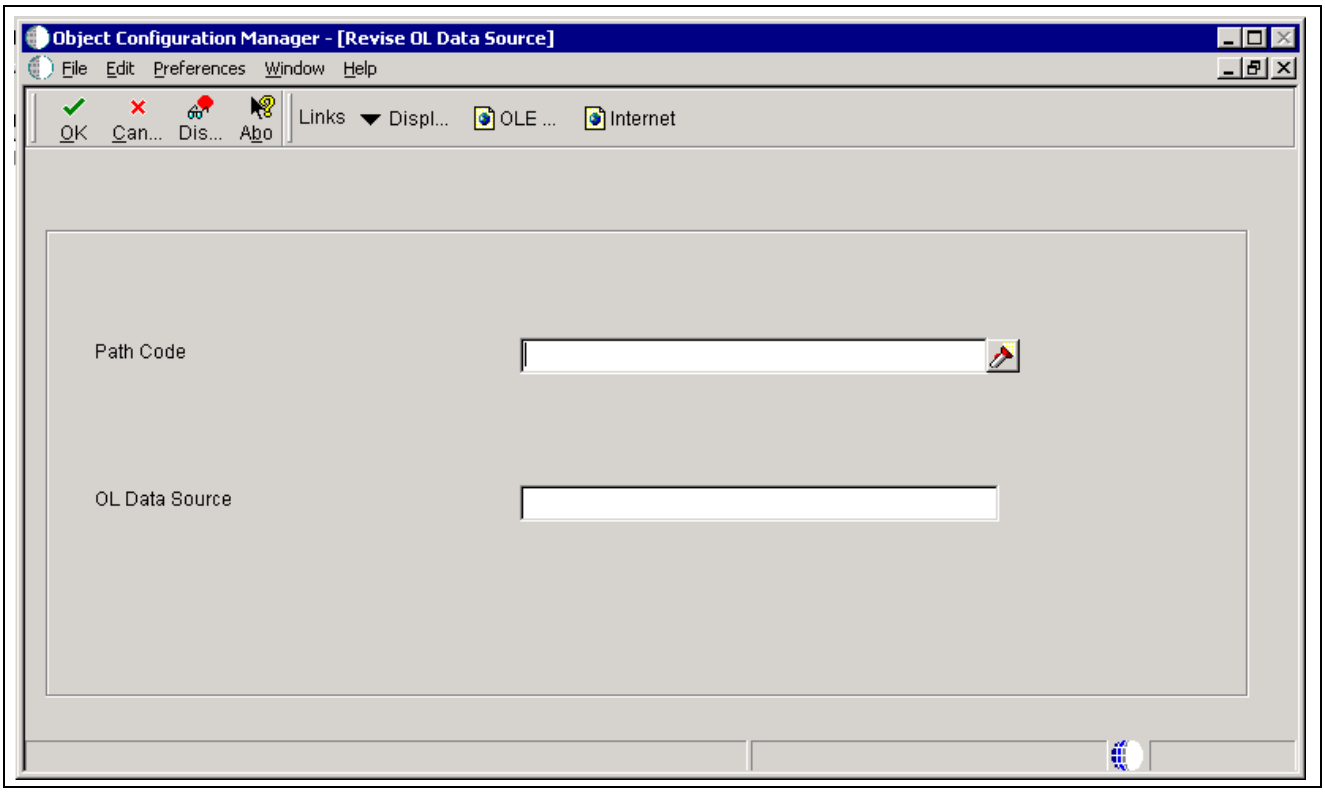
Machine Search & Select form

Machine Name A value that defines the logic host to JD Edwards EnterpriseOne.

Data Source The name that identifies the data source.

Revise OL Data Source

Access the Revise OL Data Source form.



Revise OL Data Source form

Path Code Specify the path code to a set of JD Edwards EnterpriseOne objects that will be used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

OL Data Source Specify the name that identifies the data source.

Updating the Object Configuration System Table

This section discusses how to update the object configuration system table.

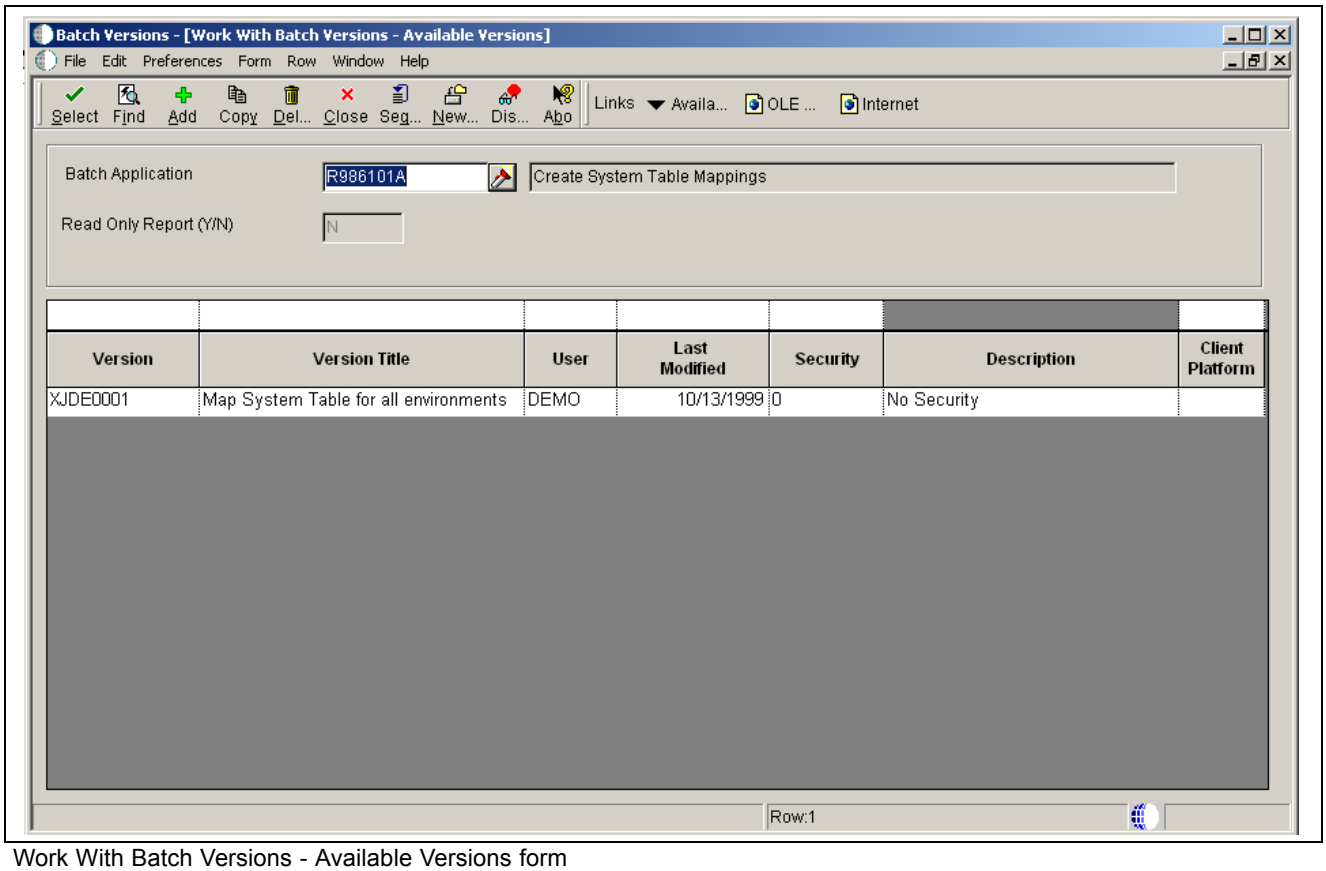
Forms Used to Run the Object Configuration System Table Update

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Data Source Management Advance and Technical Operations, Object Configuration System Table Update (P98305).	Used to add OCM records to the system table.
Versions Prompting	W98305D	In the Work With Batch Versions-Available Versions form, focus on a versionand select it.	Used to change data selection or data sequencing before submitting a report.

Running the Object Configuration System Table Update

This program adds active Object Configuration Manager (OCM) records for a specified table, user ID, and data source for all environments listed in the Environment Detail table (F00941). You can use data selection to filter the environments to which this process adds OCM records.

Access the Work With Batch Versions - Available Versions form.



Work With Batch Versions - Available Versions form

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, 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.

Version Title A description of the version that appears next to the version number. The version title is different from the report title.

This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.

User Identifies the user ID of the user that last modified the application or version.

Last Modified Indicates the last time the application or version was modified by the specified user.

Security This field allows you to restrict user access for a report version.

Values are:

0 - No security - Anyone can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

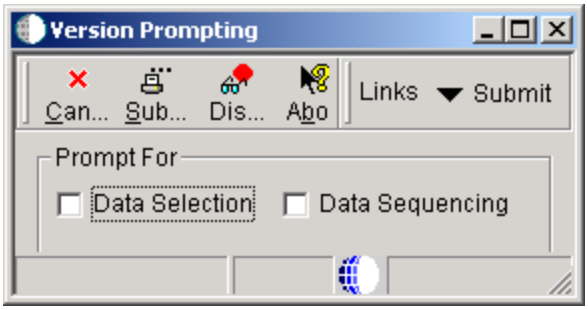
1 - Medium Security Only the "Last Modified By" user can design, change processing option values, change detail values, check in, check out, or delete the version. Anyone call install, copy, transfer, or run the version. This is how JDE Demo versions are delivered.

2 - Medium to Full Security Only the "Last Modified By" user can design, change processing option values, change detail values, check in, check out, transfer, delete or run the version. Anyone call install or copy the version.

3 - Full Security Only the "Last Modified By" user can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

Description A user defined name or remark

Client Platform A category code associated with the Versions List table for JD Edwards EnterpriseOne (F983051).



Version Prompting form

Data Selection Turn this option on to change data selection before submitting the report.

Data Sequencing Turn this option on to change data sequencing before submitting the report.

Setting Processing Options for Object Configuration System Table Update

Processing options enable you to specify the default processing for programs and reports.

Process

These processing options enable you to specify the table name, data source, and user ID to use in creating an OCM mapping. You can also specify whether to run the report in proof or final mode.

1. Enter a specific Table Name Define which mailbox a message is sent to. Values are:

Blank: The field will be not visible when creating a new message

1 The field will be visible when creating a new message

2. Enter a specific Data Source Specify the data source you want to map to each environment. The data source, along with the values you specify for the Table Name and User ID processing options, is used to create a whole OCM Mapping.

3. Enter a specific User ID Identify the user ID that is mapped to each environment. The user ID, along with the values you specify for the Table Name and Data Source processing options, is used to create an OCM mapping.

4. Proof / Final Mode Indicate whether records are changed or not changed when the batch application is run. Values are:

Blank: Proof mode

1 Final mode

Creating OCM Records for Business Functions

This section provides and overview of OCM Business Functions and discusses how to run the Create Server Business Function OCM Records Report.

Understanding Create OCM Records for Business Functions

This batch process (R986140) reads the Object Librarian tables for server business functions, and then creates Object Configuration Manager records for those business functions in the target data source that you specify in processing options. Processing options also enable you to specify the source data source and environment to use when creating these Object Configuration Manager records.

Forms Used to Create OCM Records for Business Functions

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Data Source Management Advance and Technical Operations, Create OCM Records for Business Functions(P98305).	Used to create OCM records for business functions.

Creating OCM Records for Business Functions

Access Work With Batch Versions-Available Versions form.

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, 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.

Version Title A description of the version that appears next to the version number. The version title is different from the report title.

This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.

User Identifies the user ID of the user that last modified the application or version.

Last Modified Indicates the last time the application or version was modified by the specified user.

Security This field allows you to restrict user access for a report version.

Values are:

0 - No security - Anyone can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

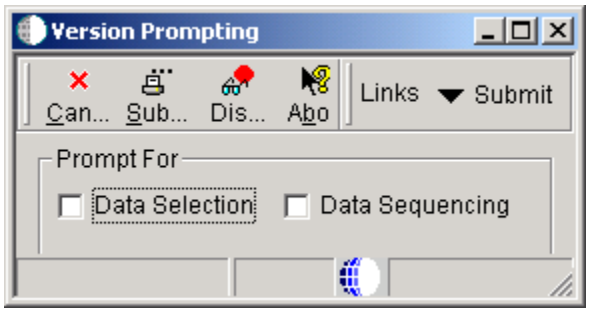
1 - Medium Security Only the “Last Modified By” user can design, change processing option values, change detail values, check in, check out, or delete the version. Anyone call install, copy, transfer, or run the version. This is how JDE Demo versions are delivered.

2 - Medium to Full Security Only the “Last Modified By” user can design, change processing option values, change detail values, check in, check out, transfer, delete or run the version. Anyone call install or copy the version.

3 - Full Security Only the “Last Modified By” user can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

Description A user defined name or remark

Client Platform A category code associated with the Versions List table for JD Edwards EnterpriseOne (F983051).



Version Prompting form

Data Selection Turn this option on to change data selection before submitting the report.

Data Sequencing Turn this option on to change data sequencing before submitting the report.

Setting Processing Options for Create OCM Records for Business Functions

Processing options enable you to specify the default processing for programs and reports.

Process

These processing options enable you to specify the run mode, data source, environment and target data source to use in creating OCM records for business functions.

1. Specify proof or final mode. When you enter 1, the report runs in Final mode, which means that the report will be printed and reports will be updated. When you enter 0 or leave this field blank, the report runs in Proof mode, which means that the report will be printed but not updated.

2. Enter the Data Source Name. Specify the machine data source that corresponds to the server you wish to run the business function on.

3. Enter the Environment Name. Specify the environment you wish to use realtime events on.

4. Enter the name of the Target Data Source where the OCM records are to be placed. Specify the system data source that the client-to-server machines use.

Updating the Oracle Parameters Table

This section discusses how to update the Oracle parameters table.

Forms Used to Update the Oracle Parameters Table

Form Name	FormID	Navigation	Usage
Work With Oracle Database Object Sizing	W9861151A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Data Source Management, Object Configuration Manager (P986110). In the Machine Search & Select form, highlight the row you want and click Select. In the Work With Object Mappings form, highlight the row you want and then select Oracle Parmsfrom the Row menu.	Used to modify the Oracle database object table fields.
Oracle Database Object Sizing	W9861151B	In the Work With Oracle Database Object Sizing form, click Add.	Used to revise and add table and index information for a data source or table.
Revise Database and Data Source Overrides	W9861151C	In the Work With Oracle Database Object Sizing form, select Overrides from the Formmenu.	Used to select override information for an Oracle table.

Updating the Oracle Parameters Table

You must update the Oracle parameters table if you use Oracle and do not follow the recommended naming conventions or if you add new Oracle data sources.

Machine Search & Select

Machine Search & Select is used to display the data source that stores the Object Configuration Manager table. The data source resides on a machine. Workstations use the system data source for their mappings. Each enterprise server has its own server map data source.

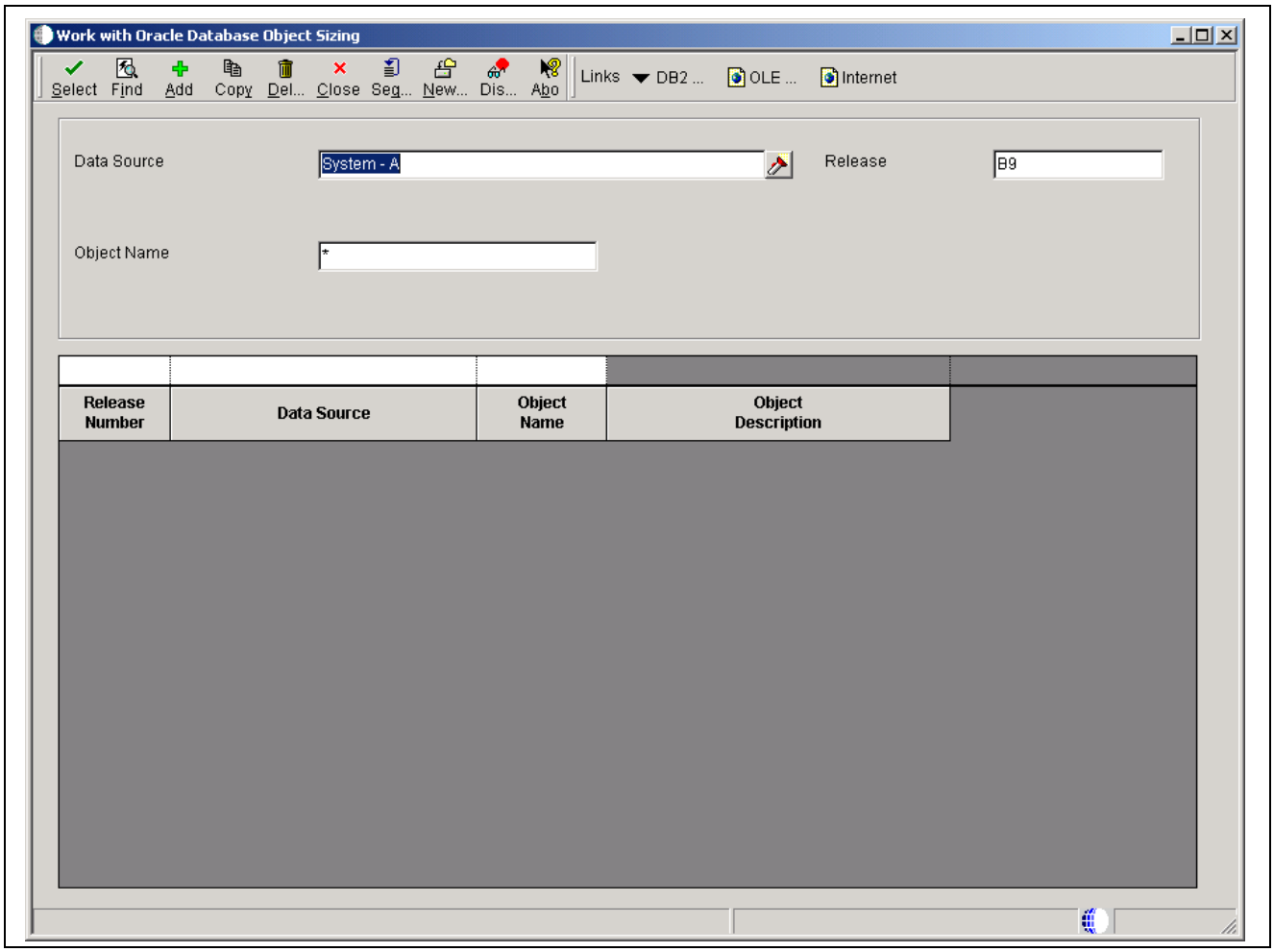
Access the Machine Search & Select form.

Machine Name A value that defines the logic host to JD Edwards EnterpriseOne.

Data Source The name that identifies the data source.

Work With Oracle Database Object Sizing

Access the Work With Oracle Database Object Sizing form.



Work With Oracle Database Object Sizing form

Release Number Specify the release number as defined in the Release Master.

Depending on whether you select a data source for the DEFAULT map or for a specific object, different fields appear on the Oracle Database Object Sizing form.

For the DEFAULT map, you can define only the name of the Oracle space where tables or indexes reside. However, for a specific object you can define parameters such as the amount of space to use for an Oracle table.

Data Source Specify the name that identifies the data source.

Object Name Define parameters for all objects in a data source, type *DEFAULT* in this field.

Enter 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. Examples of system objects include:

- Batch Applications (such as reports)
- Interactive Applications
- Business Views

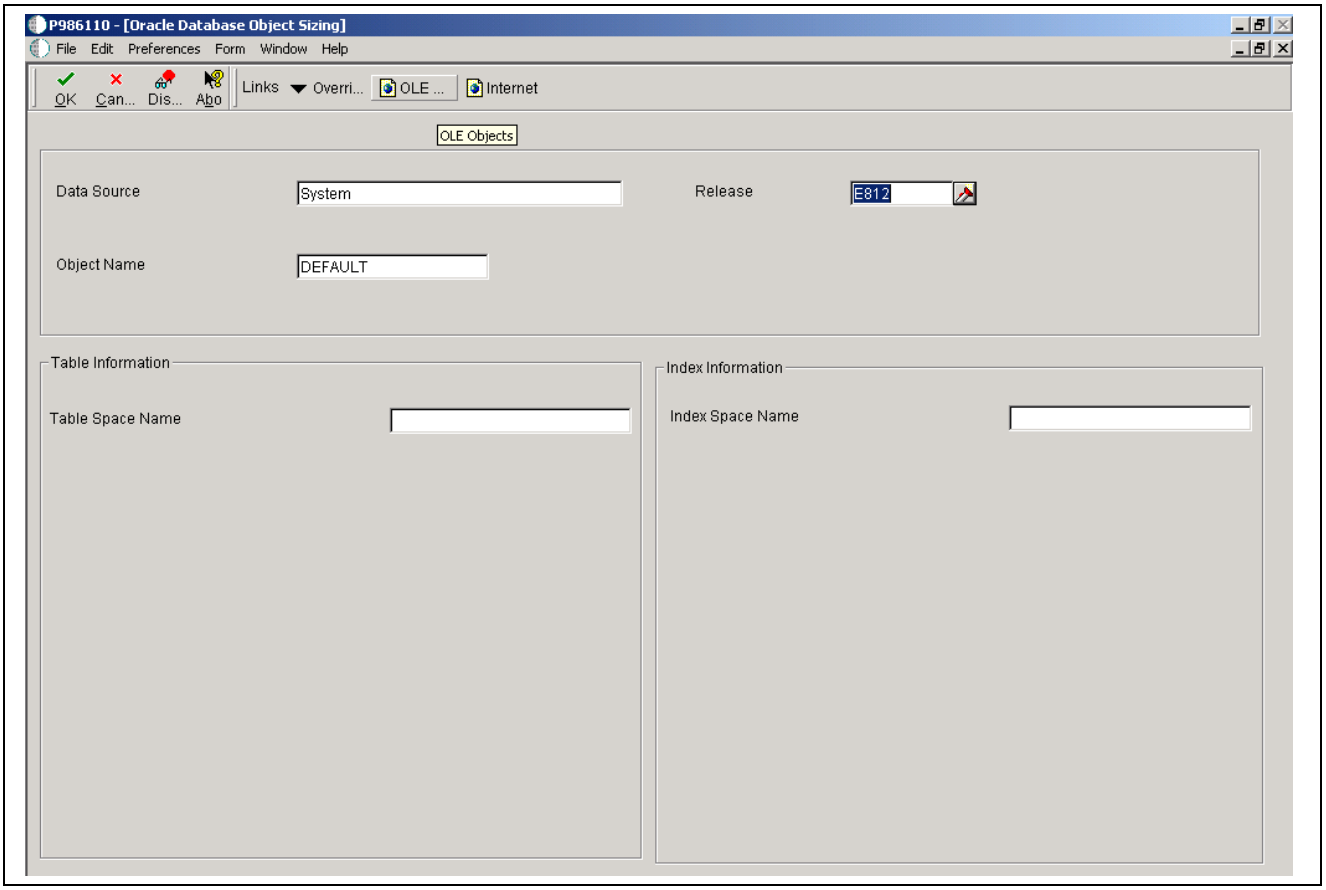
- Business Functions
- Business Functions Data Structures
- Event Rules
- Media Object Data Structures

Object Description Specify a user defined name or remark.

Oracle Database Object Sizing

Oracle Database Object Sizing is used to revise and add table and index information for a data source or table.

Access the Oracle Database Sizing form.



Oracle Database Object Sizing form

Data Source Specify the name that identifies data source.

Object Name Specify 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. Examples of system objects include:

- Batch Applications (such as reports)
- Interactive Applications
- Business Views

- Business Functions
- Business Functions Data Structures
- Event Rules
- Media Object Data Structures

Release Specify the release number as defined in the Release Master.

Depending on whether you select a data source for the DEFAULT map or for a specific object, different fields appear on the Oracle Database Object Sizing form.

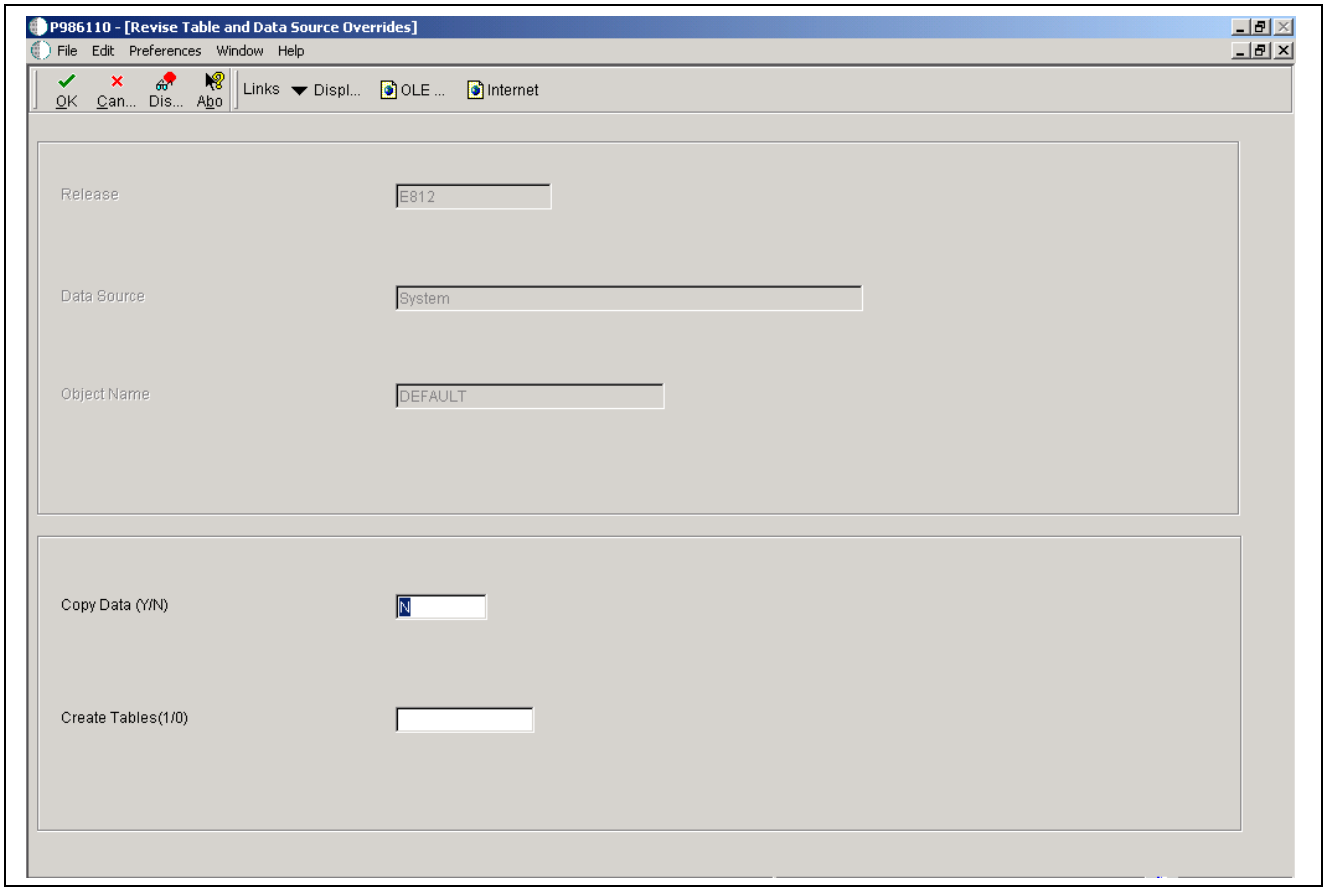
For the DEFAULT map, you can define only the name of the Oracle space where tables or indexes reside. However, for a specific object you can define parameters such as the amount of space to use for an Oracle table.

Table Space Name Specify the name of an Oracle region of space created to store tables.

Index Space Name Specify the name of an Oracle region of space created to store indexes.

Revise Table and Data Source Overrides

Access the Revise Table and Data Source Overrides form.



Revise Table and Data Source Overrides form

Copy Data (Y/N) Indicate if a file and its data are copied into production. A value of *N* moves the file without data during a file copy. When the system creates a

production data library from JDFDATA, this field designates whether the data is included in the copy.

Create Tables (1/0) Denote whether tables are automatically created for this data source.

See UDC H96/CR.

Revising the Generic Text Language Status Table

This section discusses how to revise the Generic Text Language Status table.

Understanding Revising the Generic Text Language Status Table

The Generic Text Language Status table (F001651) has more mapping flexibility than other objects because the data stored in this table has different uses. The generic text categories are as follows:

Common data that all environments use. For example, data dictionary glossaries and business function notes are the same across all environments.

Production data specific to an environment. For example, inventory item notes and address book supplemental data can be different for a corporation running JD Edwards EnterpriseOne over multiple environments, such as production, test, and demo data.

Example: Object Configuration Master Table (F986101)

This table is an example what the Object Configuration Master table (F986101) looks like after installing JD Edwards EnterpriseOne if your business follows the typical configuration: (This example includes only the PD900 (production) and TS900 (test) environments.)

Environment	Object Name	Description	Data Source	Object Type
PD900	F00165	Generic Text Table	Business Data - Prod	TBLE
PD900	GT92002	Data Dictionary Glossary	Data Dictionary	GT
PD900	GT9860A	Object Librarian	Object Librarian	GT
PD900	GT9862A	Business Function Notes	Object Librarian	GT
PD900	GT98DSA	Data Structure Notes-Structure	Object Librarian	GT
PD900	GT98DSB	Data Structure Notes-Structure and Item	Object Librarian	GT
PD900	GT98TMPL	Media Objects Templates	Object Librarian	GT
TS900	F00165	Generic Text Table	Business Data - Test	TBLE

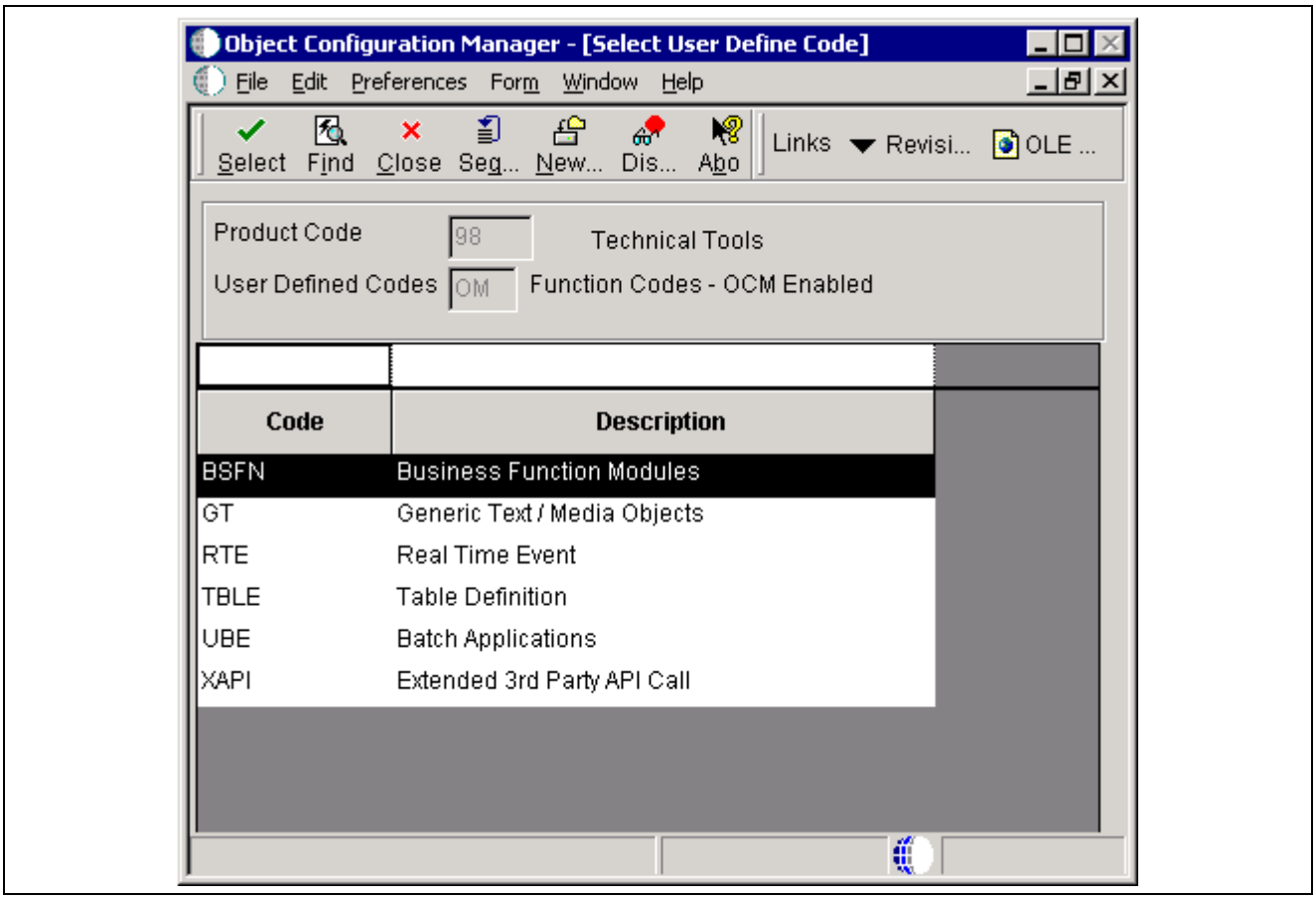
Environment	Object Name	Description	Data Source	Object Type
TS900	GT92002	Data Dictionary Glossary	Data Dictionary	GT
TS900	GT9860A	Object Librarian	Object Librarian	GT
TS900	GT9862A	Business Function Notes	Object Librarian	GT
TS900	GT98DSA	Data Structure Notes-Structure	Object Librarian	GT
TS900	GT98DSB	Data Structure Notes-Structure and Item	Object Librarian	GT
TS900	GT98TMPL	Media Objects Templates	Object Librarian	GT

Forms Used for Revising the Generic Text Language Status Table

Form Name	FormID	Navigation	Usage
Select User Defined Code	W0005SB	In the Work With Object Mappings form, click Add. In the Object Type field, click Search and Select.	Used to map specific generic text objects.

Revising the Generic Text Language Status Table

Access the Select User Defined Code form.



Select User Defined Code form

Code A list of valid codes for a specific user defined code list.

Description A user defined name or remark.

CHAPTER 11

Running Object Configuration Management Reports

This chapter provides and overview of Oracle's JD Edwards Object Configuration Management reports and discusses how to:

- Run the Job Master Deletion by Days Old report
- Run the Object Configuration Mapping Comparison report
- Run the Object Configuration Global Update report
- Run the Object Configuration Delete report
- Run the Object Configuration Copy report
- Run the OCM Category Update/Delete report

Understanding Object Configuration Management Reports

This section provides an overview of Object Configuration Management reports and discusses how to use a batch application to set processing options for running the reports.

Understanding Object Configuration Management Reports

Batch Versions provide batch processes that produce reports that helps you track your object mappings.

Some of these reports are called when you select that report from the Oracle JD Edwards EnterpriseOne menu while other reports are called from a batch application. Although the result is the same, you set processing options and data selection differently for a batch process than for a report called from a batch application.

Any data selection for a report must be entered for the specific report and not from the batch application. When you finish entering processing options and data selection, if any, you can run the report from the batch application.

- Job Master Deletion By Days Old
- Object Configuration Mapping Comparison
- Object Configuration Global Update
- Object Configuration Delete
- Object Configuration Copy
- OCM Category Update/Delete report

Running Job Master Deletion by Days Old Report

This section lists the prerequisites and discusses how to set up and run the Job Master Deletion by Days Old report.

Understanding the Job Master Deletion by Days Old Report

This batch application lets you produce a report listing obsolete print jobs submitted to servers. You have the option of generating the report only, or generating the report and then deleting obsolete records from the Job Control Status Master table (F986110).

This batch application is launched by a report driver. Enter any data selection from the batch application, but enter processing options from the report driver (R9861101), not from the actual application (R9861102).

Prerequisites

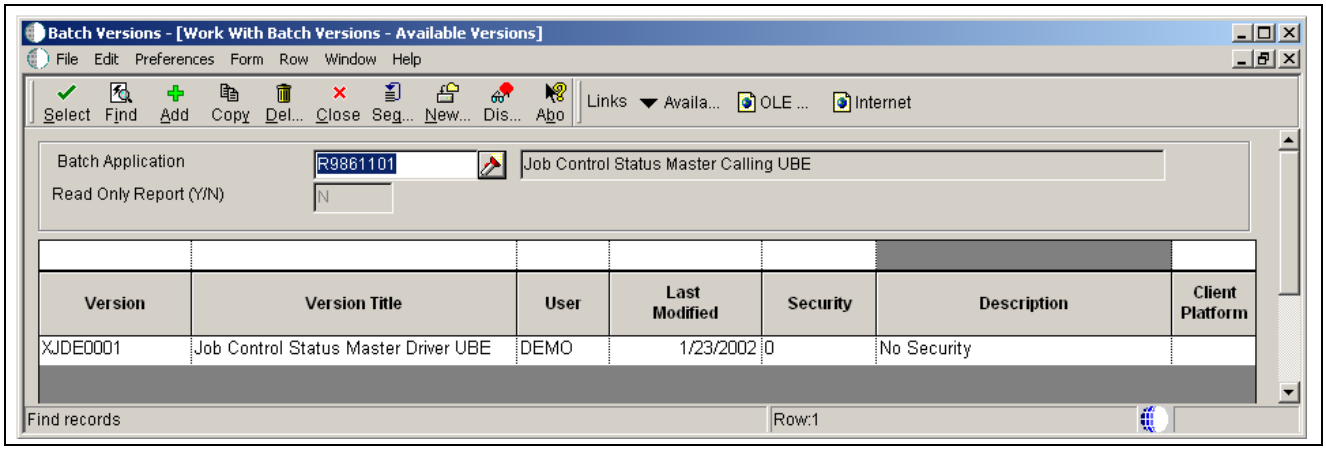
Set up processing options for the report from Batch Versions, not from the actual report.

Forms Used by Job Master Deletion by Days Old Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Report Management, Report Management Advanced and Technical Operations, Job Master Deletion by Days Old Report(P98305).	Used to produce a report listing obsolete print jobs submitted to servers.

Running the Job Master Deletion by Days Old Report

Access the Work With Batch Versions - Available Versions form.



Work With Batch Versions - Available Versions form

Versions 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 batch jobs or reports. To run a batch process, you must choose a version.

Version Title A description of the version that appears next to the version number. The version title is different from the report title.

This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.

User Identifies the use ID of the user who last modified the application or version.

Last Modified Indicates the last time an application or version was modified by the specified user.

Security This field allows you to restrict user access for a report version. Values are:

0 - No security. Anyone can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is the default when adding a new version.

1 - Medium security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version. This is how JDE Demo versions are delivered.

2 - Medium to full security. Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version.

3 - Only the "Last Modified By" user can design, change processing values, change detail values, check in, check out, install, transfer, copy, delete, or run the version.

4 - Medium security-extended. Only the "Last Modified By" user can design, change processing values (including runtime processing options and data selection), change detail values, check in, check out, transfer, delete, or run the version. Anyone can install or copy the version.

Description A user defined name or remark.

Client Platform A category code associated with the Versions List table for JD Edwards EnterpriseOne (F983051).

Setting Processing Options for Job Master Deletion by Days Old Report

Use these processing options to set up the defaults and versions for the report.

Defaults

Use this processing option to set up the data source, number of days to query for, and the report type.

1. Data Source Specify the name that identifies the data source.

2. Days Old Specify the number of days in future to query for responses due.

3. Control Mode Select an option that specifies the type of processing for an event.

1 = Proof Mode

2 = Final Mode

4. Delete Mode Specify the records to delete.

1 = Delete both UBE and Report Definition records

2 = Delete UBE records only

3 = Delete Report Definition records only

Versions

Use this processing option to specify a specific version.

1. Job Control Status UBE (R9861102) Select a user defined specification that determines how application and reports run. You use versions to group and save 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.

XJDE0001 = Jobs With All Status

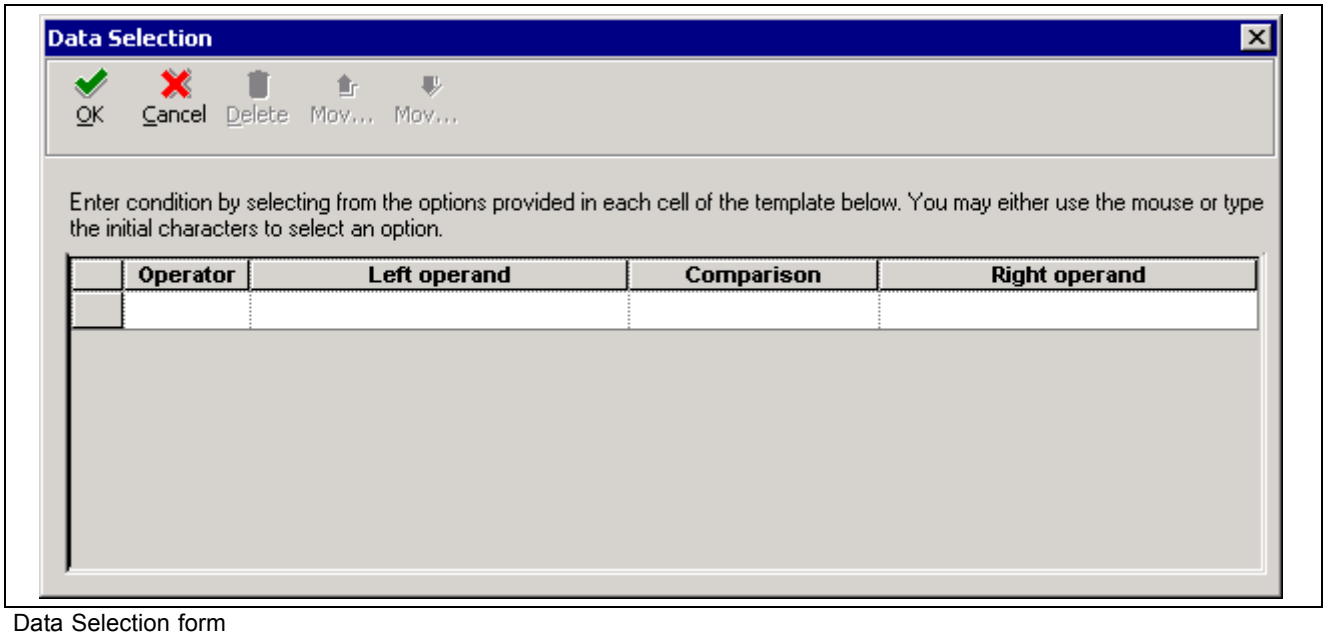
XJDE0002 = Jobs With Done Status

XJDE0003 = Jobs With Error Status

XJDE0004 = Jobs With Wait Status

Setting Data Selection for the Job Master Deletion by Days Old Report

Access the Data Selection form.



Data Selection form

Enter condition by selecting from the options provided in each cell of the template.

Operator Where

Left Operand Select the primary data source from the list.

Comparison	is equal to is equal to or empty is greater than is greater than or equal to is less than is less than or equal to is not equal to
Right Operand	Select the secondary data source from the list.

Running Interactive and Batch Applications

This section lists the prerequisites and discusses how to set up and run Interactive and Batch Applications.

Understanding Interactive and Batch Applications

These batch applications have processing options that are entered from Batch Applications (R98611A) using the same processing option form:

- Object Configuration Mapping Comparison (R986101)
- Object Configuration Global Update (R986110)
- Object Configuration Delete (R986120)
- Object Configuration Copy (R986121)

Prerequisites

Set up processing options for the report from Batch Versions, not from the actual report.

Forms Used for Object Configuration Batch Applications

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305).	Used to set up and run batch applications and reports.

Setting Processing Options for Batch Applications

These processing options enable you set other parameters used by the specific batch application.

Driver

Use these processing options to specify which report and version to run.

1. Object Configuration Manager Report Name Specify the name of the report that you want to run. Values are:

R986101 = OCM Mapping Comparison

R986110 = OCM Global Update

R986120= Object Configuration Delete

R986121 = Object Configuration Copy

2. Version Specify the version of the report that you want to run.

You must enter report processing options before running the report.

If data selection is necessary, it must be done on the version of the report that you specified on the Driver tab.

R986101

Use these processing options to specify parameters when running the R986101 report.

1. Data Source One Specify the names of the data sources that you want to compare. If necessary, you can use the Visual Assist feature to locate valid data sources. Click the Visual Assist button to display the Data Source Search and Select form.

2. Comparison Data Source Two Specify the names of the data sources that you want to compare. If necessary, you can use the Visual Assist feature to locate valid data sources. Click the Visual Assist button to display the Data Source Search and Select form.

3. Comparison Method Specify the comparison method used. Values are:

1 Compare one direction only. Print only the records found in the first data source, but not in the second data source.

Blank: Compare both directions. Print records found in the first data source but not in the second data source, and also the records in the second data source that don't exist in the first data source.

4. Exceptions Specify whether to print only the report exceptions, or every record from the data selection, noting the differences between the data sources. Values are:

1 Print exceptions only

Blank: Print all records

R986110

Use these processing options to specify parameters when running the R986110 report.

1. Process Mode Specify the mode in which the report processes data. Valid options are:

1 Proof Mode. This mode produces a report that enables you to view the records you want to delete, but no records are deleted.

Blank: Final Mode. This mode produces a report of records you want to delete, and then deletes the records.

2. Data Source name Specify the name of the data source for the Object Manager Configuration table. Any updates that occur as a result of running this report will be made to the Object Manager Configuration table found in the specified data source.

3. Process Control

Specify whether to create new OCM records or change existing ones. Values are:

1 Create new OCM records from existing ones. The records created will look similar to those selected, except that the new OCM values entered in the processing options will be substituted where appropriate.

Blank: Change selected records with new OCM values. Use data selection to specify which records will be changed with the new OCM values entered into the processing options.

4. Enter the new OCM values:

Use this processing option to enter new OCM values. If you are creating new OCM records from existing ones, any values you enter for this processing option will replace the existing ones. Values are:

Environment Name

Object Name

Primary Data Source

User

Object Type

Data Source Mode

Secondary Data Source

Allow QBE

R986120

Use these processing options to specify parameters when running the R986120 report.

1. Process Mode

Specify the mode in which the report processes data. Valid options are:

1 Proof Mode. This mode produces a report that enables you to view the records you want to delete, but no records are deleted.

Blank: Final Mode. This mode produces a report of records you want to delete, and then deletes the records.

2. Enter Data Source name

Specify the name of the data source for the Object Manager Configuration table. Any updates that occur as a result of running this report will be made to the Object Manager Configuration table found in the specified data source.

3. Object Copy

Specify whether to run the Object Configuration Copy report (R986121) along with the Object Configuration Delete report. Values are:

1 Run the Object Copy (R986121). Creates a copy before deleting records. When you run the Object Copy, remember to enter the processing options under the R986121 tab, and to set up any data selection. Data Selection should be defined on the same version of R986121 that you specified for R986121 (Object Configuration Delete).

Blank: Do not run Object Copy. Deletes the records without first creating a copy.

R986121

Use these processing options to specify parameters when running the R986121 report.

1. Process Mode Specify the mode in which the report processes data. Valid options are:

1 Proof Mode. This mode produces a report that enables you to view the records you want to delete, but no records are deleted.

Blank: Final Mode. This mode produces a report of records you want to delete, and then deletes the records.

2. From Data Source Specify the name of the data source with the Object Configuration Manager table from which you want to copy records.

3. Target Data Source Specify the name of the data source with the Object Configuration Manager table to which you want to copy.

Setting Data Selection for Object Configuration Batch Applications

Access the Data Selection form.

Enter condition by selecting from the options provided in each cell of the template.

Operator Where

Left Operand Select the primary data source from the list.

Comparison is equal to

is equal to or empty

is greater than

is greater than or equal to

is less than

is less than or equal to

is not equal to

Right Operand Select the secondary data source from the list.

Running the Object Configuration Mapping Comparison Report

This section lists prerequisites and discusses how to set up and run the Object Configuration Mapping Comparison Report.

Understanding the Object Configuration Mapping Comparison Report

Run the Object Configuration Comparison report to compare Object Configuration Master tables (F986101) from two different data sources and display the differences between them. For example, you might compare the F986101 table in the system data source to the F986101 table for a given server map data source.

Prerequisite

Set up processing options for the report from the report driver, not from the actual report.

Forms Used to Run the Object Configuration Mapping Comparison Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305). In the Batch Application field, enter *R986101*, then click Find.	Used to compare Object Configuration Master tables (F986101) from two different data sources and display the differences between them.

Running the Object Configuration Mapping Comparison Report

Access the Work With Batch Versions - Available Versions form.

Running the Object Configuration Global Update Report

This section lists prerequisites and discusses how to set up and run the Object Configuration Global Update Report.

Understanding the Object Configuration Global Update Report

This process performs global updates and global copies of object mappings in the same Object Configuration Master table (F986101). You can use this batch application rather than the interactive application (P986110). This application is useful in updating and copying multiple records.

Prerequisite

Set up processing options for the report from the report driver, not from the actual report.

Forms Used to Object Configuration Global Update Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305). In the Batch Application field, enter *R986101*, then click Find.	Used to performs global updates and global copies of object mappings in the same Object Configuration Master table (F986101).

Running the Object Configuration Global Update Report

Access the Work With Batch Versions - Available Versions form and run the Object Configuration Update Report (R986110).

Running the Object Configuration Delete Report

This section lists prerequisites and discusses how to set up and run the Object Configuration Delete Report.

Prerequisite

Set up processing options for the report from the report driver, not from the actual report.

Form Used to Run the Object Configuration Delete Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305). In the Batch Application field, enter *R986120*, then click Find.	Used to delete object mappings in the same Object Configuration Master table (F986101).

Running the Object Configuration Delete Report

Access the Work With Batch Versions - Available Versions form and run the Object Configuration Delete Report (R986120).

Running the Object Configuration Copy Report

This section lists the prerequisites and discusses how to set up and run the Object Configuration Copy Report.

Understanding the Object Configuration Copy Report

This process copies Object Configuration Manager records from one data source to another and automatically deletes any duplicate records. For example, if you create a new environment by copying an existing one, Object Configuration Manager records are created for the new environment in the system data source.

You use the Object Configuration Copy batch process to copy those records to the appropriate server map data source. After copying the records, use the Object Configuration Global Update program (R986110) to change, at the minimum, any mappings for LOCAL to the appropriate server location.

Prerequisite

Set up processing options for the report from the report driver, not from the actual report.

Forms Used for the Object Configuration Copy Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305). In the Batch Application field, enter *R986121*, then click Find.	Used to copy Object Configuration Manager records from one data source to another and automatically deletes any duplicate records.

Running the Object Configuration Copy Report

Access the Work With Batch Versions - Available Versions form and run the Object Configuration Copy Report (R986121).

Running the OCM Category Update/Delete Report

This section provides an overview of the OCM Category Update/Delete Report and discusses how to set up and run the OCM Category Update/Delete report.

Understanding the OCM Category Update/Delete Report

This batch application (R986101B) lets you add, update, or delete member objects of a specified category from the Object Configuration Master table (F986101).

Prerequisites

Set up processing options for the report from the report driver, not from the actual report.

Forms Used for the OCM Category Add/Update/Delete Report

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	In Solution Explorer, navigate to Application Development, Object Management, Interactive and Batch Versions, Batch Versions (P98305). In the Batch Application field, enter *R986101B*, then click Find.	Used to add, update, or delete member objects of a specified category from the Object Configuration Master table (F986101).

Running the OCM Category Add/Update/Delete Report

Access the Work With Batch Versions - Available Versions form and run the Add/Update/Delete Report (R986101B).

Setting Processing Options for the OCM Category Add/Update/Delete Report

Processing options for the OCM Category Add/Update/Delete report include:

OCM Modes

These processing options let you add or delete OCM mappings for all members in the same category rather than adding or deleting them individually.

Proof or Final Mode Enter *1* to run the report in Proof mode. The report will print and the Object Configuration Manager table will not be updated. Enter 2 to run the program in Final mode. The report will print and the Object Configuration Manager table will be updated.

Add or Delete Mode Enter *1* to add OCM mappings for objects in the category or 2 to delete mappings.

Override Mapping Enter *1* to delete OCM mappings for objects that already have mappings for the environment and user. Enter 2 or leave this field blank to deactivate but not delete mappings for conflicting OCM records.

OCM Settings

These processing options enable you to specify whether to add or delete mappings, as well as the appropriate path code, environment, and data source to use.

Path Code to use when adding or deleting OCM mappings When you enter a path code, mappings are added or deleted for all environments with the path code you enter. You do not need to enter a path code if you entered an environment name in the Environments processing

option. If this field is blank, OCM mappings will be added or deleted regardless of the path code.

Environment to use when adding or deleting OCM mappings

If this field is blank, OCM mappings are added or deleted regardless of the environment. If you entered a path code in the Path Code processing option, that path code is used. If both the Environments and Path Code processing option fields are blank, OCM mappings will be added or deleted for all environments in the Environment Detail table (F00941).

User Class/Role

If you are deleting OCM mappings and this field is blank, all OCM records will be deleted regardless of the user class or role. If you are adding OCM mappings, you must enter a value in this field.

Data Source

If you are deleting OCM mappings and this field is blank, all OCM records will be deleted regardless of the data source. If you are adding OCM mappings, you must enter a value in this field.

CHAPTER 12

Understanding Application Communication

The chapter provides an overview of middleware used in application communication and discusses:

- JDENET communication middleware.
- JDEBase database middleware.
- Direct Connect Processing

Middleware

In a client/server environment, applications must communicate across different platforms. These platforms can have different communications protocols, database management systems, and hardware operating systems. For clients to communicate with servers and servers to communicate with other servers, a mechanism must exist that can bridge multiple protocol and multiple vendor issues. This mechanism is a layer of software called middleware, which resides between the operating system and the business applications. It is important to have an application architecture that is based on a single, consistent middleware strategy.

Oracle's JD Edwards EnterpriseOne provides these types of middleware:

Middleware	Description
JDENet Communication Middleware	Performs the connections from client to server and server to server, and sends messages for distributed requests. It is a peer-to-peer, message-based, socket-based, multi-process communication middleware solution.
JDEBase Database Middleware	Provides platform-independent application program interfaces (APIs) for multiple database access. These APIs are used in these two ways: • By JD Edwards EnterpriseOne applications that dynamically generate platform-specific Structured Query Language (SQL), depending on the data source request. • As open APIs for writing advanced business functions in the C programming language. The software uses these APIs to dynamically generate platform-specific SQL statements. JDEBase also provides client-to-server and server-to-server database access. To accomplish this, the software is integrated with a variety of third-party database drivers, such as IBM Client Access/400 database software and Microsoft Open Database Connectivity (ODBC) programming interface.

JDENet Communication Middleware

To communicate with each other across a network, the two computers must share a communications protocol (or set of protocols). A communications protocol is a formal set of rules that specifies the format and relationship for exchanging data among different devices. The communication middleware is concerned with these protocol layers:

Middleware	Protocol Layers
Network Layer	The network layer handles addressing and routing information, error checking, and retransmission requests.
Transport Layer	The transport layer provides connection-oriented data-delivery services across networks. This layer provides end-to-end data exchanges in which systems maintain a session or connection with each other for the reliable, sequenced exchange of data. JD Edwards EnterpriseOne supports the TCP/IP protocol suite.
Application Layer	The application layer provides application-to-application interaction and data exchange. JDENet is the application layer communication middleware.

JDENet Communication Middleware

JDENet is the JD Edwards EnterpriseOne proprietary communication middleware package that provides server-to-server and client-to-server communication.

JDENet is a peer-to-peer middleware solution. For example, think of a client as a network conversation initiator and a server as a network conversation responder. In this example, a client always initiates the conversation by asking for something from another machine, and a machine acts as a server when it responds to a network request, such as when it gives something asked for by the client. In this peer-to-peer middleware solution, the distinction between client and server is determined by which machine starts the conversation. Any machine, running on any platform, can act as a client or as a server at a given time.

With JDENet, communication between client and server occurs through messages. JDENet messages contain processing requests, such as requests for business functions, batch jobs, or JD Edwards EnterpriseOne login security. JDENet messages can originate from the client or the server. JDENet handles database requests only if multiple servers are in use and if they are different server types.

Application requests (messages) can be synchronous or asynchronous. A synchronous message, such as calling a business function, requires the client to wait for the server to complete the request. An asynchronous request, such as a batch process, enables the client to continue with another task while the software processes the request. In some circumstances, business functions can also be called asynchronously.

Socket-Based Communication

A socket is a communications endpoint through which an application sends or receives packets of data across a network. Sockets provide a duplex communication channel between one or more systems. JDENet uses stream sockets to provide end-to-end communications. Sockets guarantee that the data arrives intact.

Message-Based Communication

Message-based communication means that applications send service requests for logic or data in the form of messages that are received and stored in a queue for processing. The middleware handles message transmission, which enables the client application to process other tasks. Without messaging services to handle these jobs, the application must wait until the request is handled and the results returned.

Messaging is most appropriate for event-driven applications. It is the opposite of remote procedure calls (RPC), which are synchronous. The message packaging and "handshaking" of JDENet ensures that the message transmission is complete.

Process-Based Design

Although client workstations can have more than one copy of JD Edwards EnterpriseOne loaded, only one JD Edwards EnterpriseOne Windows-executable application can be running at any one time. The software uses an internal network process (also referred to as a net process) called JDENet to communicate a request to the JD Edwards EnterpriseOne server.

Servers also have a net process called JDENet. This process communicates with the client workstations and routes request messages to appropriate dedicated JD Edwards EnterpriseOne processes. In turn, the dedicated processes route work to the appropriate platform-specific logic processes, such as DLLs, shared libraries, and job queues. A server can have multiple JD Edwards EnterpriseOne main processes, multiple dedicated processes, multiple DLLs, shared libraries, and job queues.

The advantage of this architecture is that multiple workstations can make requests to the same server at one time. You can control the number of workstations that can make and maintain a session connection to a main server process. You also can define the total number of dedicated processes (and the number of each type) that the software uses to process specific types of workstation requests.

Network Processes

A relationship exists among network processes, dedicated processes, and logic processes. This relationship is specifically defined by the jde.ini file on the enterprise server. Every enterprise server must have at least one JD Edwards EnterpriseOne network process, which is referred to as a JDENET_n job. This job handles network connections and traffic for JD Edwards EnterpriseOne.

As defined in the jde.ini file for each server, multiple JDENET_n processes can exist. Regardless of the number of JDENET_n processes that exist, the initial JDENET_n process serves as the master listener. On a Windows server this master listener is called JDESNET.

If multiple JDENET_n jobs are specified, the software starts the jobs as required, allocating a job to each request. When the maximum number of JDENET_n processes is started, the software automatically alternates between the currently running JDENET_n jobs until the maximum number of connections is reached, providing load balancing among network processes. If the maximum number of connections for the JDENET_n job is met on a given server, a client or server cannot initiate an additional JD Edwards EnterpriseOne session on that server until an existing session connection is ended. By design, all connections to JDENET_n persist for the duration of a session.

For example, suppose that the jde.ini file on the server specifies that four JDENET_n processes are enabled. The first JDENET_n request is routed to the master listener, which is the initial JDENET_n process that is run at server start-up. When a second request to JDENET_n is received, the master listener receives the request and assigns it to a second JDENET_n process, which it then starts. This assignment persists for the duration of the session between the requesting device and the server. The same process occurs for the third and fourth JDENET_n requests. When the fifth request is received, it is assigned to the first JDENET_n process, and the cycle continues.

Kernel Processes

The JDENET_n process is responsible for handling the network layer of communication. If the JDENET_n job determines that the incoming message is a request for logic processing, it routes the request to an appropriate JDENET_k job. The software determines an appropriate JDENET_k job based on message identifiers. The JDENET_k job is the process that provides the link between the JDENET_n job and the appropriate platform-specific processing job. The JDENET_k process is applicable only to servers.

The JDENET_k job handles the two-way routing to and from the various logic processes, and the JDENET_n job handles the return delivery to the appropriate machine. Many dedicated kernel types exist, and each is responsible for a specific type of JD Edwards EnterpriseOne process.

Examples of logic processes include dynamic link libraries (.dll) for Windows platforms, shared libraries (.sl or .so) for UNIX platforms, and JDENet processes for iSeries platforms.

JDEBase Database Middleware

Different database management systems (DBMS) have their own version of Structured Query Language (SQL). For example, this demonstrates how Microsoft SQL Server, Oracle, and DB2/400 handle the same SQL statement:

DBMS	SQL Statement
Microsoft SQL Server	SELECT * FROM PRODDTA.F0101

DBMS	SQL Statement
Oracle	SELECT * FROM PRODDTA.F0101;
DB2/400	SELECT * FROM PRODDTA/F0101

The purpose of a database middleware layer is to provide a common interface to interpret the various versions of SQL. JD Edwards EnterpriseOne has a database middleware product called JDEBase, which is a common set of application programming interfaces (APIs) that programmers can call to request data and perform data manipulation logic. JDEBase interprets the generic APIs and converts the SQL into the appropriate statements for JD Edwards EnterpriseOne to access the database.

Multiple databases in a distributed environment require a monitoring program to ensure database integrity. This monitoring program is referred to as a transaction monitor. The JDEBase database middleware has an embedded transaction monitor.

JDEBase provides:

- The ability to insulate developers from platform-specific SQL coding
- Rapid development of native drivers
- Server-to-server communication
- Transaction processing

JDEBase provides a set of APIs to the developer and a set of translation programs to JD Edwards EnterpriseOne. The translation programs are embedded in the data source definitions.

For example, suppose a data request for Address Master is made. The Object Configuration Manager (OCM) determines which data source contains the requested table. The Data Source Master table (F98611) provides the database information.

Working with Direct-Connect Processing

This section provides overviews recommendations for Data and Logic Distribution and discusses how to:

- Set up direct-connect processing
- Set up object mapping for direct-connect environments

Understanding Recommendations for Data and Logic Distribution

To achieve the best performance with direct-connect processing, use these recommendations for data and logic distribution:

- Map the transaction data to a data server.
- Map user defined codes and menus to the workstation. You also can map other static files locally. If the maintenance costs are more than the performance returns in mapping these tables locally, you can map them to a data server.
- Map all batch applications to the enterprise logic server.

Setting Up Direct-Connect Processing

In direct-connect processing, workstations are connected directly to servers that can store data and process logic. For performance reasons, distribute the data and logic in a manner that reduces network traffic and unnecessary input and output on the server.

Before performing the tasks in this section, you will need to have all workstations connected directly to servers.

Setting Up Direct-Connect Processing

The process overview for setting up direct-connect processing is as follows:

- Create a production environment and verify that the new environment uses a production path code.
- Determine the name of the master business functions that you should map to the server, if applicable.
- Modify the Object Configuration Manager mappings for the new environment.

See Also

Appendix B, "Setting Up Environments Manually," Adding an Environment, page 162

Chapter 10, "Working with Object Configuration Manager," page 95

Setting Up Object Mapping for Direct-Connect Environments

This section lists the prerequisites and discusses how to set up a master data administration environment.

Before performing the tasks in this section, you will need to:

Locating Master Business Functions	In a direct-connect environment, you need to identify the master business functions so that you can map them to a server.

Setting Up Master Data Administration Environment

Depending on the configuration, you might require multiple direct-connect environments. For example, to support multi-tiered configurations you might need an environment that maps all application processing to one server and all data to a corporate server.

The data administrator uses the Master Data Administration environment to maintain the published tables in the central location.

To set up a Master Data Administration environment:

- Map all table objects to a business data source on the server.
- Map user defined codes and menus to a control table data source on the server.

  The control data contains the published set of user defined codes.
- Map system tables to the system data source.
- Map Object Librarian tables to the Object Librarian data source.
- Map data dictionary tables to the data dictionary data source.
- Map batch applications to the server.

CHAPTER 13

Understanding Typical Customer Configurations

This chapter provides an overview of customer configurations and discusses:

- Recommended setup
- Configuration data

Recommended Configurations

Oracle's JD Edwards EnterpriseOne can be configured in many ways. Examples and recommendations for the setup follow.

You should follow JD Edwards EnterpriseOne-recommended setup and naming standards wherever possible, unless a strong business case exists to support the need to change. Following the typical setup and naming standards enhances the likelihood of success and minimizes confusion when communicating with individuals outside the core project team who are not aware of your specific configuration.

If you want to customize your configuration, you should change only the descriptions with the typical setup, not the names. Upgrades will be easier with fewer manual steps if you use the JD Edwards EnterpriseOne-recommended naming standards.

An environment description is important because the description appears on the environment list of the login screen where the user selects the environment. The environment description should define:

- Path code
- Data type (such as production, test, or prototype)
- Data location
- Location that batch applications will execute

Basic Environments

You should understand the basic environments of a JD Edwards EnterpriseOne configuration. For each environment, this chapter describes:

- The path code
- The purpose
- The object mappings

For a given release, all environments you receive from JD Edwards EnterpriseOne share common data sources. For example, for JD Edwards EnterpriseOne 8.11, these data sources are common to all environments:

- System - 900

- Object Librarian - 900
- Data Dictionary - 900

Environment-Specific Data Sources for UNIX and Windows

These data sources are specific to a particular UNIX and Windows environment. For each environment, a separate data source must exist.

- Central objects
- Versions tables
- Business data
- Control tables
- Local - *xxxx*

  Where *xxxx* is the path code for the environment.

Production Environment (PD900)

This table explains the data sources for the production environment.

Data Source	Environment
Path Code	PD900
Central Objects data source	Central Objects - PD900
Versions Tables data source	Versions - PD900
Business Data data source	Business Data - PROD
Next Numbers data source	Control Tables - Prod
Tasks/UDCs data source	Control Tables - Prod

Prototype Environment (PY900)

This table defines the data sources for the prototype environment.

Data Source	Environment
Path Code	PY900
Central Objects data source	Central Objects - PY900
Versions Tables data source	Versions - PY900
Business Data data source	Business Data - CRP
Next Numbers data source	Control Tables - CRP
Tasks/UDCs data source	Control Tables - CRP

Development Environment (DV900)

This table defines the data sources for the development environment.

Data Source	Environment
Path Code	DV900
Central Objects data source	Central Objects - DV900
Versions Tables data source	Versions - DV900
Business Data data source	Business Data - TEST
Next Numbers data source	Control Tables - Test
Tasks/UDCs data source	Control Tables - Test

Pristine Environment (PS900)

This table defines the data sources for the pristine environment.

Data Source	Environment
Path Code	PS900
Central Objects data source	Central Objects - PS900
Versions Tables data source	Versions - PS900
Business Data data source	Business Data - PS900
Next Numbers data source	Business Data - PS900
Tasks/UDCs data source	System Local - PS900

Planner Environment (PSFTPLAN)

This table defines the data sources for the deployment environment.

Data Source	Environment
Path Code	PLANNER
Central Objects data source	Not applicable
Planner Tables data source	Planner - 900
Versions Tables data source	Versions Local
Business Data data source	Business Data Local

Data Source	Environment
Next Numbers data source	Control Tables Local
Tasks/UDCs data source	Control Tables Local

Deployment Environment (DEP900)

This table defines the data sources for the deployment environment.

Data Source	Environment
Path Code	PLANNER
Central Objects data source	Not applicable
Versions Tables data source	Versions Local
Business Data data source	Business Data Local
Next Numbers data source	Control Tables Local
Tasks/UDCs data source	Control Tables Local

Remote Environments

Remote environment names are preceded by a 3-character location code followed by a J for Java application server.

For example:

*xxx*JPD900 indicates a Java application server for the PD900 environment at location *xxx*.

The path codes and data sources for remote environments are identical to the base environments.

PD900 Environment

This table defines the data sources for a remote production environment.

Data Source	Environment
Path Code	PD900
Purpose	PD900 is the live production environment for the end users who will have a tested and released package on their machine. Batch applications run on the server. Eventually, more than one production environment might be established for different types of distributed data, logic, and modes of processing.

PD900 Object Mappings

This table explains the object mappings for a production environment.

Mapping	Explanation
Business Data - PROD	The default object mapping. Tables that are not specifically mapped by other data sources use this data source as their default mapping. This mapping includes the F00165 - Media Objects Storage table.
Central Objects - PD900	Maps to the central object tables, including F98950 - User Overrides.
Control Tables - Prod	Maps to the next number tables.
Data Dictionary - 900	Maps to the data dictionary tables and the data dictionary media object text in the GT92002 - Data Dictionary - Glossary Information data structure.
Logic Data Source	Maps to the machine on which batch applications are run.
Object Librarian - 900	Maps to the object librarian tables and the object librarian media object text in these data structures: • GT9860A - Object Librarian Generic Text Structure • GT9862A - Business Function Notes • GT98DSA - Data Structure Notes - Structure and Item • GT98DSB - Data Structure Notes - Structure and Item • GT98TMPL - Media Object Templates
Local - PD900	Maps to the user defined code tables.
System - 900	Maps to the system tables.
Versions - PD900	Maps to the version tables, including: • F983051 - Versions List • F98306 - Processing Option Text

PY900 Environment

This table defines the data sources for a remote prototype environment.

Data Source	Environment
Path Code	PY900
Purpose	PY900, the prototype environment, is the staging environment for production. Constants tables and master tables (such as company constants, fiscal date patterns, and item master) are populated with customer data during the prototype process. Copy the tables to the production environment before you go live. After you run the Installation Workbench, no business data exists until you enter it. When appropriate, you should refresh the test data from PD900, which represents the production data.

PY900 Object Mappings

This table explains the object mappings for a prototype environment.

Mapping	Explanation
Business Data - CRP	The default object mapping. Tables that are not specifically mapped by other data sources will use this data source as their default mapping. This mapping includes the F00165 - Media Objects Storage table
Central Objects - PY900	Maps to the central object tables, including F98950 - User Overrides.
Control Tables - CRP	Maps to the next number tables.
Data Dictionary - 900	Maps to the data dictionary tables and the data dictionary media object text in the GT92002 - Data Dictionary - Glossary Information data structure.
Logic Data Source	Maps to the machine on which batch applications are run.
Object Librarian - 900	Maps to the object librarian tables and the object librarian media object text in these data structures: • GT9860A - Object Librarian Generic Text Structure • GT9862A - Business Function Notes • GT98DSA - Data Structure Notes - Structure and Item • GT98DSB - Data Structure Notes - Structure and Item • GT98TMPL - Media Object Templates
Local - PY900	Maps to the user defined code tables.

Mapping	Explanation
System - 900	Maps to the system tables.
Versions - PY900	Maps to the version tables including: • F983051 - Versions List • F98306 - Processing Option Text

DV900 Environment

This table defines the data sources for a remote development environment.

Data Source	Environment
Path Code	DV900
Purpose	DV900, the development environment, is the testing environment for development objects. This environment shares the test data that TS900 uses. Developers log in to this environment to modify objects and test them before transferring the changed objects to the PD900 path code. Once you have transferred objects into PD900, a user can install a recent PRD package that has not been released to end users and log in to either CRP or TST for additional testing.

DV900 Object Mappings

This table defines the object mappings for a development environment.

Mapping	Explanation
Business Data - PROD	The default object mapping. Tables that are not specifically mapped by other data sources use this data source as their default mapping. This mapping includes the F00165 - Media Objects Storage table
Central Objects - DV900	Maps to the central object tables, including the F98950 - User Overrides table.
Control Tables - Production	Maps to the next number tables.
Data Dictionary - 900	Maps to the data dictionary tables and to the data dictionary media object text in the GT92002 - Data Dictionary - Glossary Information data structures.
Logic Data Source	Maps to the machine on which batch applications are run.

Mapping	Explanation
Object Librarian - 900	Maps to the object librarian tables and to the object librarian media object text in these data structures: • GT9860A - Object Librarian Generic Text Structure • GT9862A - Business Function Notes • GT98DSA - Data Structure Notes - Structure and Item • GT98DSB - Data Structure Notes - Structure and Item • GT98TMPL - Media Object Templates
Local - DV900	Maps to the user defined code tables.
System - 900	Maps to the system tables.
Versions - DV900	Maps to the version tables, including; • F983051 - Versions List • F98306 - Processing Option Text

PS900 Environment

This table illustrates the data sources for a remote test environment.

Data Source	Environment
Path Code	PS900
Purpose	Use the PS900 environment to test pristine (unaltered) objects with JD Edwards EnterpriseOne demonstration data. Also use this environment for training classes. You must have this environment to compare modified objects to pristine objects. When you encounter a software problem that JD Edwards EnterpriseOne Worldwide Customer Support cannot duplicate, they will ask you to log in to the pristine environment to duplicate the problem. Routinely (such as monthly or quarterly), you should refresh the data that this environment uses with the JD Edwards EnterpriseOne demonstration data shipped with the software.

PS900 Object Mappings

This table illustrates the object mappings for a test environment.

Mapping	Explanation
Business Data - PS900	The default object mapping. Tables that are not specifically mapped by other data sources use this data source as their default mapping. This mapping includes the F00165 - Media Objects Storage table.

Mapping	Explanation
Data Dictionary - 900	Maps to the data dictionary tables and to the data dictionary media object text in the GT92002 - Data Dictionary - Glossary Information data structure.
Logic Data Source	Maps to the machine on which batch applications are run.
Object Librarian - 900	Maps to the object librarian tables and to the object librarian media object text in these data structures: • GT9860A - Object Librarian Generic Text Structure • GT9862A - Business Function Notes • GT98DSA - Data Structure Notes - Structure and Item • GT98DSB - Data Structure Notes - Structure and Item • GT98TMPL - Media Object Templates
Local - PS900	Maps to the user defined code tables.
System - 900	Maps to the system tables.
Versions - PS900	Maps to the version tables, including: • F983051 - Versions List • F98306 - Processing Option Text

Data Sources

If the enterprise server has Oracle Server or Microsoft SQL Server and is a more powerful machine than the deployment server, for performance reasons we recommend that you put all data sources (except the local data sources) on the enterprise server instead of the deployment server.

Data Sources

This table illustrates the data sources and their purpose.

Data Source Name	Owner and Purpose
Business Data - CRP	The owner is CRPDTA. The library is CRPDTA. The CRP business data. Before going live, you should copy much of this data to Business Data - PROD.
Business Data - PS900	The owner is PS900DTA. The library is PS900DTA. The pristine data shipped with the software.

Data Source Name	Owner and Purpose
Business Data - PROD	The owner is PRODDTA. The library is PRODDTA. The production business data.
Business Data - TEST	The owner is TESTDTA. The library is TESTDTA. The test data entered during CRP or converted from non-JD Edwards EnterpriseOne systems.
Central Objects - PY900	The owner is PY900. The library is COPY900. The central objects data source associated with the PY900 path code. After you create and test the modifications in this path code, transfer them to the PD900 path code.
Central Objects - DV900	The owner is DV900. The library is CODV900. The central objects data source associated with the DV900 path code. After you create and test the modifications in this path code, transfer them to the PD900 path code.
Central Objects - PS900	The owner is PS900. The library is COPS900. The central objects data source associated with the PS900 path code.
Central Objects - PD900	The owner is PD900. The library is COPD900. The central objects data source associated with the PD900 path code. Transfer objects into this data source after you have tested them in path code DV900.
Control Tables - CRP	The owner is CRPCTL. The library is CRPCTL. The control tables used in the CRP environment.
Control Tables - Prod	The owner is PRODCTL. The library is PRODCTL. The control tables used in the production environment.

Data Source Name	Owner and Purpose
Control Tables - PS900	The owner is PS900CTL. The library is PS900CTL. The control tables used in the PS900 path code.
Control Tables - Test	The owner is TESTCTL. The library is TESTCTL. The control tables used in the test environment.
Data Dictionary - 900	The owner is DD900. The library is DD900. The single data dictionary that all environments use.
LOCAL	No owner exists. The data source that defines the local machine to JD Edwards EnterpriseOne. Use it to override reports to the workstation.
machine name	No owner exists. A value that defines the logic host to JD Edwards EnterpriseOne.
Machine Name - Server Map	The owner is SVM900. The server map for the logic server.
Object Librarian - 900	The owner is OL900. The Object Librarian, which is release specific.
Control Tables Local	No owner exists. The data found in the PSCTL900 MSDE. This data is used only in the planner.
System Local	No owner exists. The data found in the PSSY900 MSDE. This data is used only in the planner.
Data Dictionary Local	No owner exists. The data found in the PSDD900 MSDE. This data is used only in the planner.
Object Librarian Local	No owner exists. The data found in the PSOL900 MSDE. This data is used only in the planner.

Data Source Name	Owner and Purpose
Versions Local	No owner exists. The data found in the PSVL900 MSDE. This data is used only in the planner.
Local - PY900	No owner exists. The source that contains the User Defined Codes (UDCs): Control Tables - CRP.
Local - DV900	No owner exists. The source that contains the User Defined Codes (UDCs): Control Tables - Test.
Local - PS900	No owner exists. The source that contains the User Defined Codes (UDCs): Business Data - PS900.
Local - PD900	No owner exists. The source that contains User Defined Codes (UDCs): Control Tables - Production.
System - 900	The owner is SY900. The library is SY900. The single set of system tables that all environments use.
Versions - PY900	The owner is PY900. The library is COPY900. The source that contains the versions list and processing option text tables for the PY900 environment.
Versions - DV900	The owner is DV900. The library is CODV900. The source that contains the versions list and processing option text tables for the DV900 environment.
Versions - PS900	The owner is PS900. The library is PS900. The source that contains the versions list and processing option text tables for the PS900 environment.
Versions - PD900	The owner is PD900. The owner is COPD900. The source that contains the versions list and processing option text tables for the PD900 environment.

Configuration Data

This table summarizes the data, environments, central objects (path codes), and packages needed:

Requirement	Explanation
Data	You might have these sets of data: • Conference Room Pilot (CRP) business data • Conference Room Pilot (CRP) control tables • Data Dictionary (all environments share) • JD Edwards EnterpriseOne pristine data • Production business data • Production control tables (used by the PD900 environment) • Object Librarian (all environments share) • System (technical data all environments share) • Test business data • Test control tables (used by TS900 and DV900 environments) • Versions
Central Objects (Path Codes)	You should have these sets of central objects or path codes, which are release-specific: • PY900 central objects • DV900 central objects • PS900 central objects • PD900 central objects

Requirement	Explanation
Environment	The installation process defines the environments, which are as follows: PY900 CRP objects (possibly testing a package you have not released to production users), with data mapped to CRP. The path code is PY900 with table objects mapped to CRP data. DV900 Development objects with test data. The path code is DV900, with table objects mapped to test data. PS900 Pristine objects with JD Edwards EnterpriseOne pristine demo data. The path code is PS900, with objects mapped to JD Edwards EnterpriseOne pristine data PD900 Production objects with production data. The path code is PD900, with table objects mapped to production data. TS900 CRP objects with test data. The path code is PY900, with table objects mapped to test data
	If you are not planning any development projects, you need only two path codes: PY900 and PD900. You should create a development path code if you plan to do extensive software modification. The fewer path codes you use, the better. With each additional path code comes version control maintenance that is time consuming unless a good reason exists for the additional path code. Even when making extensive software modifications, you should have only these four path codes (sets of central objects): PY900 This path code contains a practice set of objects that are tested during conference room pilot before transferring objects to production. It is for deploying quick fixes or making minor modifications that you will quickly transfer to production. It can also be used as a place to test modifications that were done in the development path code before taking the risk of transferring them to the production path code.

Requirement	Explanation
	DV900 Use this path code for normal development. Upon successful testing, transfer the objects to the PY900 path code, using Object Transfer, and distribute to the users through a package build and a workstation installation. PS900 This is the set of pristine objects shipped from JD Edwards EnterpriseOne. You should not make changes to this path code other than paper fixes from JD Edwards EnterpriseOne. This path code is used to compare JD Edwards EnterpriseOne standard software to any custom solutions you have implemented in other path codes. You should keep a copy of this path code so that you have a clean copy of the software in case you need to refresh anything. PD900 This is the production path code. Just-in-time installations come directly from this location, and production server objects are also deployed from here. After testing software changes in PY900, transfer them to PD900 and then deploy the changes to the enterprise servers and workstations.
Packages	All path codes share the same Object Librarian tables, the same system data source, and, normally, the same data dictionary. The only distinct tables across path codes are central objects/specifications (F987*), Versions List (F983051), and Processing Option Text (F98306). At JD Edwards EnterpriseOne. we have determined that each package should have an A and B version, and that you alternate between these versions when you build packages. If you are using both full and partial packages, you would have four packages for each path code. This setup gives you two full packages (A and B) for production and two partial packages (A and B) for production. For example: • PD900FA (Standard Production Full A) • PD900FB (Standard Production Full B) • PD900PA (Standard Production Partial A) • PD900PB (Standard Production Partial B)

APPENDIX A

Troubleshooting Business Function Processing Problems

This section provides an overview of business function processing problems and standard troubleshooting processes.

Business Function Processing Problems

The Oracle JD Edwards EnterpriseOne configurable network computing (CNC) solution enables developers and administrators to map business functions to one or more application servers for logic processing. When a problem occurs on the server, the software attempts to reconnect to the application server so that the business function can run. If the software can reconnect to the server and run the business function, work proceeds uninterrupted.

However, these circumstances can complicate business function processing:

- The client workstation cannot reconnect to the application server because a server process has died.
- Business function processing creates cache, or state information, on the application server whose process has died.
- The business function causes one or more processes to die on the server.
- The client workstation cannot reconnect to the application server because the server machine has gone down and the server machine is part of a server cluster.

When the client workstation cannot communicate with the server, the software redirects business function processing to a secondary server. A list in the CallObject code designates the name of the original server and the name of the secondary server to which future calls should be rerouted.

Note. The default configuration is that no secondary server is defined during the JD Edwards EnterpriseOne installation process. Defining a server will require changes to the OCM mappings. If you do not define a secondary server and failover occurs, the software remaps business function processing from the failed server to the client workstation.

When business function processing creates cache on the application server where a process has died, the client workstation reconnects to the application server, but the user must exit the application and restart it.

When a business function causes one or more processes to die on the server, the client workstation reconnects to the server. Because the business function is causing the jdenet_k process to die, JD Edwards EnterpriseOne fails the business function call.

When the client workstation cannot communicate with a server in a server cluster, the software recognizes that the server is part of a cluster and continues to try to reconnect. The transfer of control from one server in a cluster to another server in a cluster can take several minutes.

The JD Edwards EnterpriseOne Configurable Network Computing solution provides a methodology that handles business function failure and enables you to continue working, even when a server has failed or a kernel process has died, ending the processing of logic on an application server. In addition, the software writes a message to the jde.log whenever a failover occurs, enabling you to troubleshoot the problem.

Failure to Connect to the Server

The mechanism by which a business function fails to connect to a server depends on how the server is configured in the network. Failures for these two types of configurations are discussed in this section:

- Failure to connect to a server in a non-clustered server configuration
- Failure to connect to a server in a clustered configuration

Failure to Connect to the Server in a Non-Clustered Server Configuration

In a non-clustered server configuration, the software redirects business function processing if it cannot connect to the primary server. These steps describe what occurs during the initial stages of an attempt to call a business function to run on an application server:

1. The user calls a business function on a server.
2. The software checks to see if the server has been failed over from the primary server to a secondary server or to the client workstation.
3. If processing has been directed to another server, the software remaps the business function and sends the CallObject message to the secondary server or to the client workstation to run the business function.
4. If the server has not been failed over, the software sends the CallObject message to the original server to run the business function.

In the second phase of business function processing, the software attempts to run the logic on the application server or client workstation. These steps describe what occurs during the second stage of processing:

1. If the business function runs without error, either on the original server or the failover alternative, the request has been processed.
2. If the client workstation request is not successfully processed by the server, the software increments a reconnect counter and attempts one reconnection.
3. If the value on the reconnect counter is greater than 1, the business function fails. If the value on the reconnect counter is not greater than 1, the software reconnects to the server and attempts to run the business function.
4. If the client is unable to reconnect to the server, the request is redirected to a secondary server if one is defined, or to the client workstation if one is not defined.

If cache has been created on the server, the user must exit the application and restart it.

Failure to Connect to a Server in a Clustered Configuration

If a business function fails because of a server failure in a clustered configuration, rather than failing over to a secondary server or the client workstation, the client will wait until a new machine in the cluster is available then resubmit the business function request. While trying to reconnect, the software displays a transient window: This window refreshes once a minute and continues to display until the client is able to successfully reconnect to the clustered server.

If the business function cache was created on the first server before it went down, the software will not submit the business function request to the server cluster. In this case, you must exit the application and then resubmit the business function.

Failure to Load the Business Function

When a client workstation requests to run a business function on a server, the server must successfully load the business function before it can run. This process can fail for these two reasons:

- Server cannot load the library where the business function resides.
- Server cannot get the address of the business function.

Server Cannot Load the Library Where the Business Function Resides

When the server cannot load the business function library, the software displays this message on the client workstation and writes the text of the message to the jde.log file on that machine:

```
The Business Function Library xxxx could not be loaded on
 server yyyy. Because of the unknown cache-state on the server, you must
 exit this application all the way to the menu. Please notify your
 JD Edwards EnterpriseOne System Administrator to have the problem corrected before
 attempting to run the Business Function zzzz again.
```

Probable reasons that the library failed to load are that:

- The business function library failed to build during the package build process.
- The library was inadvertently deleted or renamed.
- A problem exists with permissions.

If the library fails to load, close the application until you get to the menu, and contact your system administrator. Ensure that the problem is corrected before you attempt to re-run the business function.

Server Cannot Get the Address of the Business Function

When the server cannot get the address of the business function within the library, the software displays this message on the client workstation and writes the text of the message to the jde.log file on that machine:

```
The Business Function xxxx was not found in the Business Function
 Library yyyy on server zzzz. Because of the unknown cache-state on the
 server, you must exit this application all the way to the menu. Please
 notify your JD Edwards EnterpriseOne System Administrator to have the problem⇒
 corrected
 before attempting to run this Business Function again.
```

Probable reasons that the server cannot get the address of the business function are that:

- The package build process failed to create the module that contains the business function; therefore, the module was not included in the business function library.
- The client has a newer package than the server, and the business function exists on the client but not on the server.

If this error occurs, close the application until you get to the menu and contact your system administrator. Ensure that the problem is corrected before you attempt to re-run the business function.

Failure While the Business Function is Running

The business function itself can cause one or more processes to die on the server. In this case, the software displays a dialog box indicating that the business function is causing problems.

You might have to change OCM mappings or fix a bug in the business function if this dialog box appears.

Resetting the Server Cache

If the business function does not run the first time, the software checks to see if cache was created on the server during the first failed attempt. If no cache is created and the reconnection attempt to the primary server fails, the software attempts to run the business function on the secondary server or the client workstation.

If cache is created on the server, the software instructs the user to close the application and start over. This message is also written to the client jde.log file.

The creation of cache on the server is vital to the processing of business functions. The software creates cache when one business function runs so that one or more subsequent functions can use the data in the cache. For example, one business function might create and initialize the cache, a second might add data to it, and a third might access the data and insert it into a database.

If a process on the server dies after the first business function creates the cache and the client workstation is unable to communicate with the process on the server that contains the cache, the subsequent business functions are not able to access the original cache. Therefore, in this scenario, the software forces you to close the application and start over.

Note. UBEs and table conversions continue to process business functions after a failure, even if they create cache on the server.

APPENDIX B

Setting Up Environments Manually

This chapter provides an overview of environments and environment definitions and discusses how to work with environments.

Understanding Environments

This section provides an overview of environments and discusses:

- Environment definitions
- Environment table relationships

See Chapter 8 "Using Environment Director to Copy an Environment to a New Environment" for a more automated approach to setting up environments.

Understanding Environments

Oracle's JD Edwards EnterpriseOne environments, which you define, are collections of pointers indicating the location of data and objects. An environment definition contains a path code and a set of Oracle's JD Edwards Object Configuration Manager (OCM) mappings.

JD Edwards EnterpriseOne has these types of environments:

Understanding Environment Definitions

Environment definitions are stored in these five tables:

Library List Control Table (F0093)

The Library List Control table (F0093) contains valid environments for each user. You must assign to each user at least one valid environment for logging on to JD Edwards EnterpriseOne. The environments you assign to each user in the Library List Control table are validated at startup against the JD Edwards EnterpriseOne directories on the workstation. Only those environments for which a user is authorized and that are installed on the machine they log on to are listed as available environments.

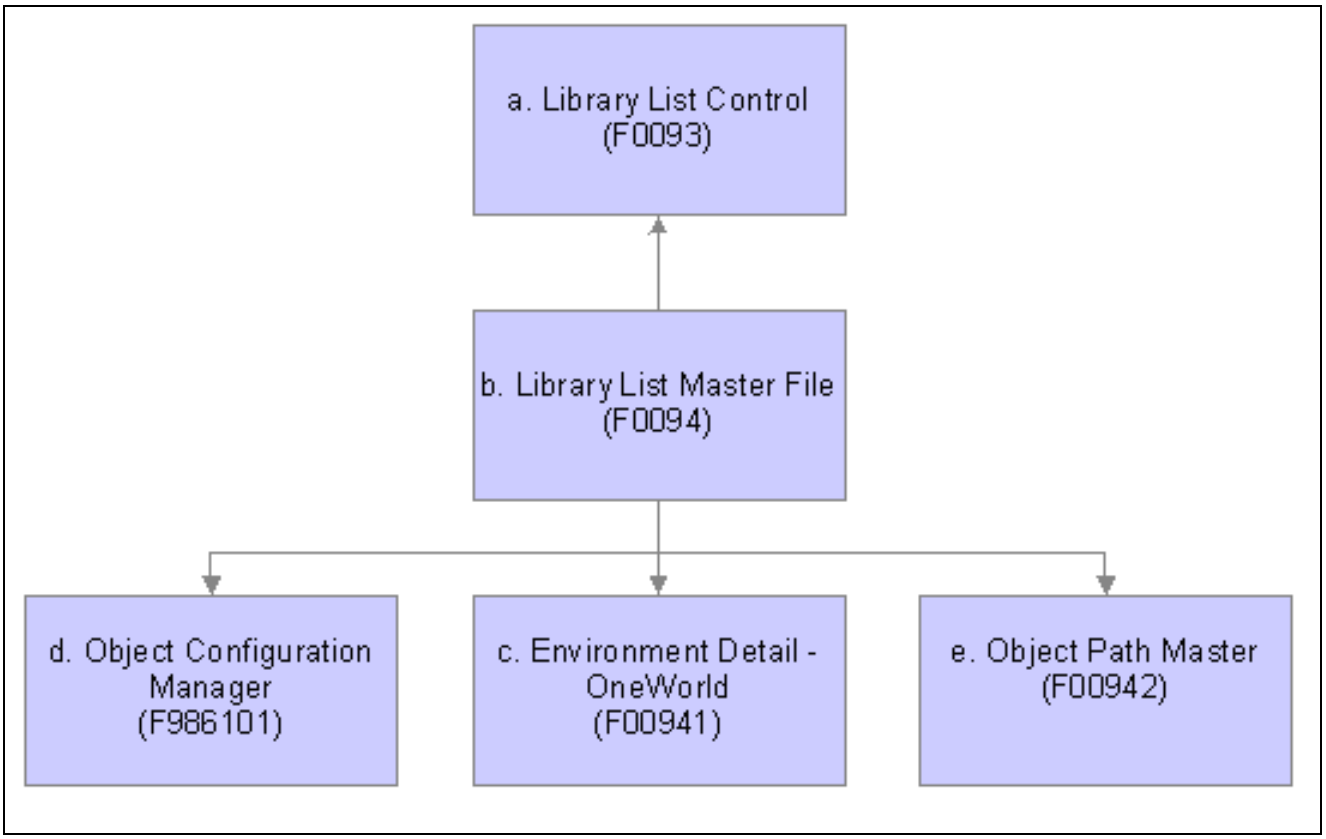
The Environment Revisions form of the User Profile application enables you to specify which users are authorized for specific environments.

Library List Master File Table (F0094)	The Library List Master File table (F0094) contains the name of the environment and the description.
Environment Detail Table (F00941)	This table contains the environment name and the associated path code from the Object Path Master File table (F00942). Every environment must have an associated path code, but environments can share the same path code. For example, two environments can use production objects and have different data location mappings, such as in the case of a group of users processing against data on a corporate server and a group of users processing against data on a departmental server.
Object Configuration Master Table (F986101)	The Object Configuration Master table (F986101) has data and logic object mappings for every environment. Every environment must have entries in the Object Configuration Master table, even if the mappings for the environments are identical. Two environments might have different path codes but have the same mappings for data and logic. For example, developers and testers could have different sets of objects, but their data is in the same database and their logic processes locally.
Object Path Master File Table (F00942)	This table stores the location of the path code's central objects, the release associated with this path code, and other details.

Understanding Environment Table Relationships

The Library List Control table and the Environment Master table have a direct correlation. For each environment listed as a valid user environment in the F0093, there is one F0094 record that provides the name and description of that environment. The F986101 and F00941 and F00942 provide additional information for each of the environments defined in the Environment Master table (F0094).

This diagram illustrates environment table relationships:



Environment Table Relationships

Setting Up Environments

This section provides and overview of setting up environments and discusses how to set up processing options using Environment Master (P0094).

Understanding Setting Up Environments

You can create a new environment either by adding a new environment or by copying an existing environment. The custom environments functionality in Installation Planner and Environment Director applications simplify this process.

Because the Object Configuration Manager (OCM) mappings are an important element of the environment, determining these mappings is the first step in deciding whether to create a new environment or copy an existing environment.

When you copy an environment, the OCM mappings for that environment are copied along with the environment. It is sensible to create a new environment by copying an existing environment when the OCM mappings for the environment you want to create closely match the mappings of an existing environment. For example, if a development environment and a test environment contain similar mappings, it might be easier to copy an environment and change the mappings that are different.

You should add a new environment rather than copying an existing environment when you do not want to use another environment's object mappings.

Note. Remember that when you add an environment without copying an existing one, you must create the OCM mappings manually. For this reason, it is typically easier to create a new environment by copying an existing one and adjusting the OCM mappings instead of creating all of them manually.

Form Used to Work with Environments

Form Name	FormID	Navigation	Usage
Work With Environments	W0094E	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094).	Used to work with environments.

Setting Processing Options for Environment Master

Processing options enable you to specify the default processing for programs and reports.

Process

These processing options give you greater flexibility when copying or deleting by enabling you to specify whether users have the ability to copy or delete OCM mappings for this environment in other data sources.

Regardless of what you enter for these processing options, the software automatically copies or deletes OCM mappings for the environment you are using. The processing option values you enter determine whether you can copy or delete mappings in other data sources.

1. Delete OCM Mappings Indicate whether users can delete Object Configuration Manager mappings for this environment from other data sources. Values are:

Blank: Users cannot delete OCM mappings.

1: Users can delete OCM mappings.

2. Copy OCM Mappings Indicate whether users can copy Object Configuration Manager records for this environment into other server map data sources. Values are:

Blank: Users can copy OCM mappings.

1: Users cannot copy OCM mappings.

Adding an Environment

This section discusses how to add an environment.

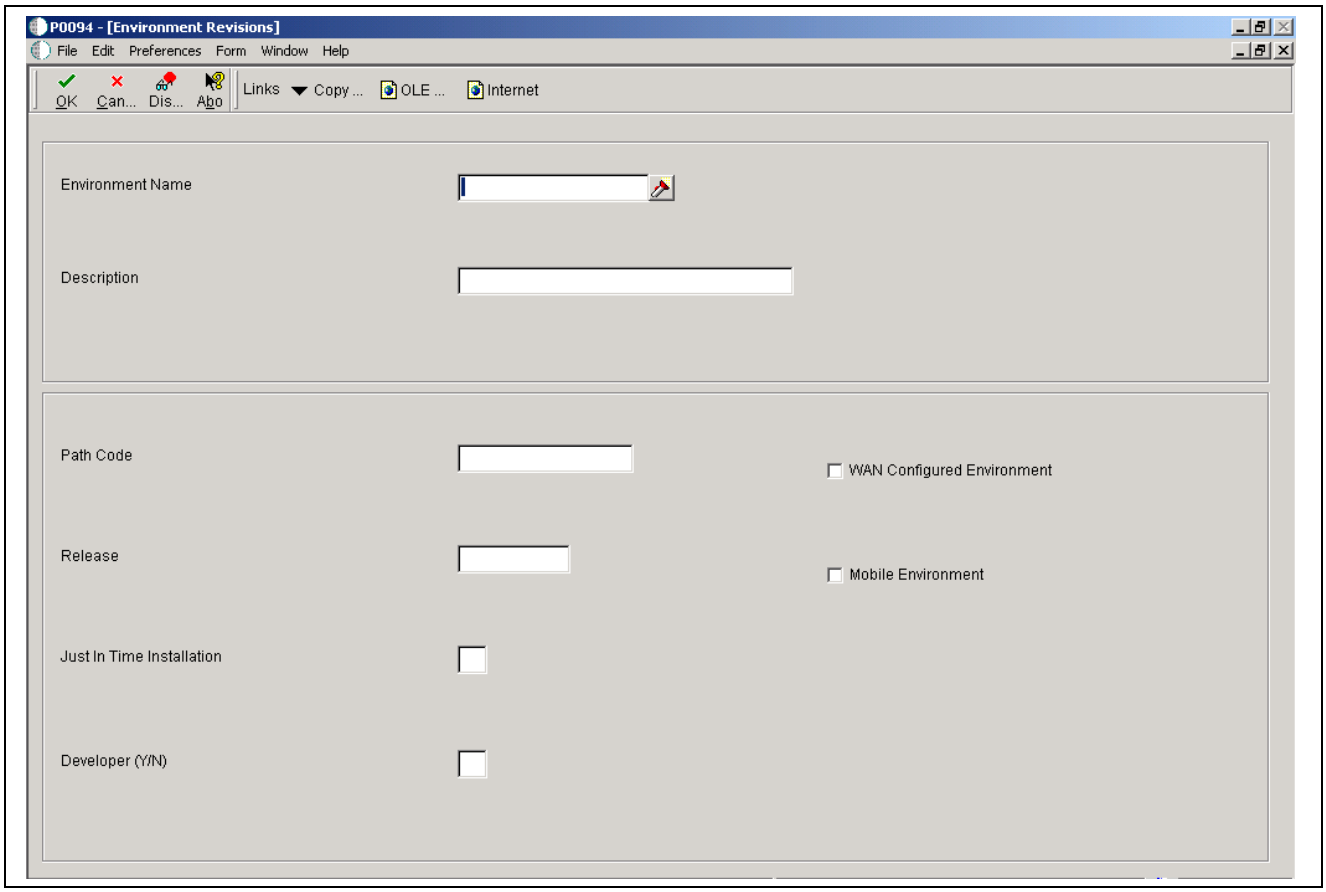
Forms Used to Add an Environment

Form Name	FormID	Navigation	Usage
Environment Revisions	W0094A	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094). In the Work With Environments form, click Add.	Used to add an environment.
Data Source Selection	W0094F	From the Environment Revisions form, the system will display the Data Source Selection form is the path code you entered does not exist in the Library Master File table (F0094).	Used to select a data source.

Adding an Environment

You should add a new environment rather than copying an existing environment when you do not want to use the OCM mappings of another environment.

Access the Environment Revisions form.



Environment Revisions form

Environment Name Specify the name associated with a specific list of libraries. The J98INITA initial program uses these library list names to control environments that a user can sign on to. These configurations of library lists are maintained in the Library List Master File table (F0094).

This field represents a valid environment that can be used to run in JD Edwards EnterpriseOne. The environment encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace within the system.

Description Add a user defined name or remark.

Path Code Specify a pointer to a set of JD Edwards EnterpriseOne objects which will be used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Release Specify the release number as defined in the Release Master.

Just In Time Installation Use this field to turn ON/OFF just-in-time installation for anyone signed onto this environment. Consider turning just-in-time installation OFF before you transfer modified applications into the production path code. Once you have fully tested the applications and are ready for production users to receive the changes, you can turn just-in-time installation back ON.

Developer (Y/N) Future Use.

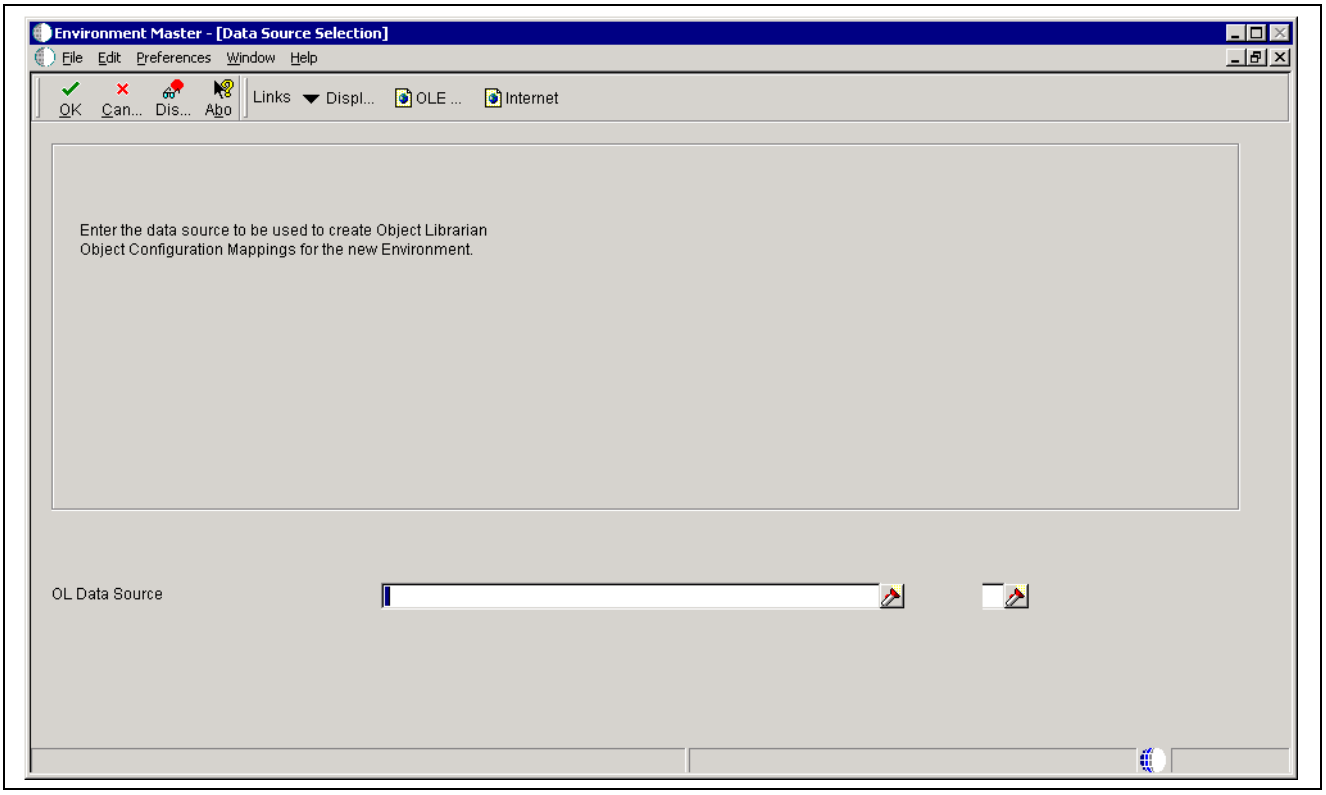
A one field for the Install group.

WAN Configured Environment Specify the code to be used

1 - WAN Configured Environment

Blank - Not Configured for WAN

Mobile Environment Specify the code the that allows the user to indicate that this environment is a mobile environment.



Data Source Selection form

OL Data Source Identify the data source.

Copying an Environment

This section discusses how to copy an environment.

Form Used to Copy an Environment

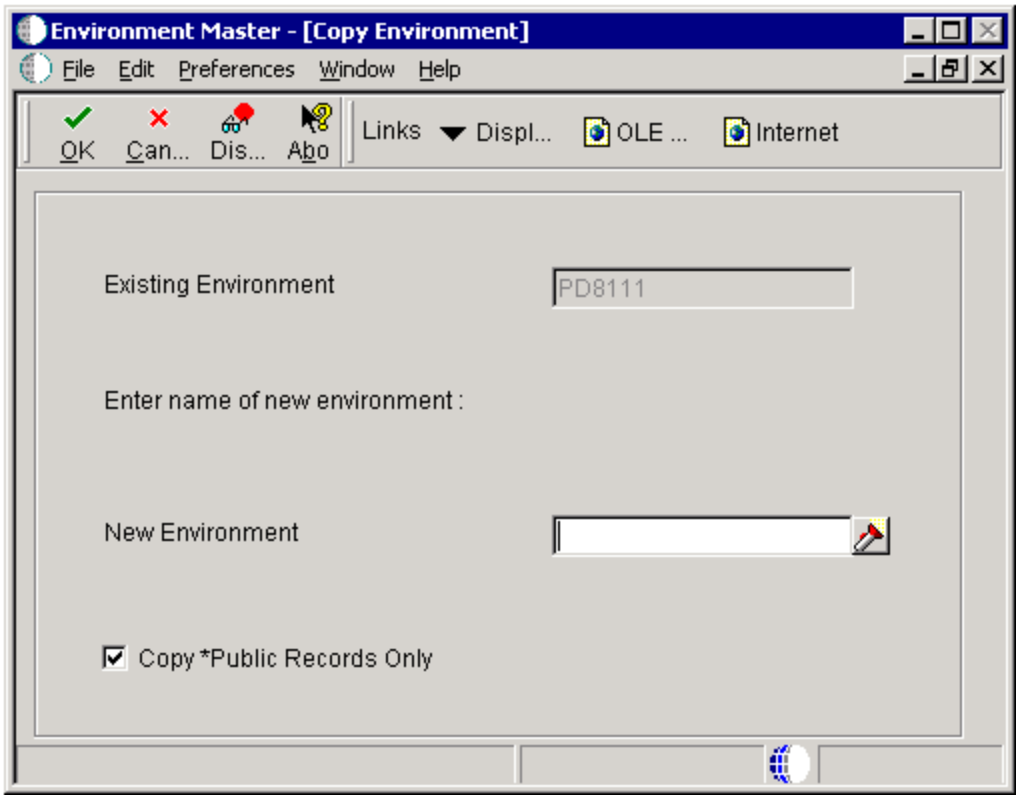
Form Name	FormID	Navigation	Usage
Copy an Environment	W0094B	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094). In the Work With Environments form, click Find. Highlight the row you want and click Select. In the Row menu, click Copy Environment.	Used to copy and environment.

Copying an Environment

When you create a new environment by copying an existing environment, the software also copies the Object Configuration Manager mappings associated with the existing environment. You can set up a processing option if you want to specify additional Object Configuration Manager mappings to copy at the time that you copy the environment.

You can copy environments from your own existing environments, or you can use the Installation Planner, which provides sample production environments you can use as templates. These templates contain suggested mappings for all JD Edwards EnterpriseOne tables.

Access the Work With Environments form.



Copy an Environment form

Existing Environment

Note. This field will be unavailable for input.

The name associated with a specific list of libraries. The J98INITA initial program uses these library list names to control environments that a user can sign on to. These configurations of library lists are maintained in the Library List Master File table (F0094).

This field represents a valid environment that can be used to run in JD Edwards EnterpriseOne. The environment encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace within the system.

New Environment

Specify the name associated with a specific list of libraries. The J98INITA initial program uses these library list names to control environments that a user can sign on to. These configurations of library lists are maintained in the Library List Master File table (F0094).

This field represents a valid environment that can be used to run in JD Edwards EnterpriseOne. The environment encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace within the system.

Copy *PUBLIC Records Only

When copying an environment, the OCM records are also copied.

Check this box to copy only the *PUBLIC OCM records to the new environment.

Copying a New Environment to a New Path Code

This section discusses how to copy a new environment to a new path code.

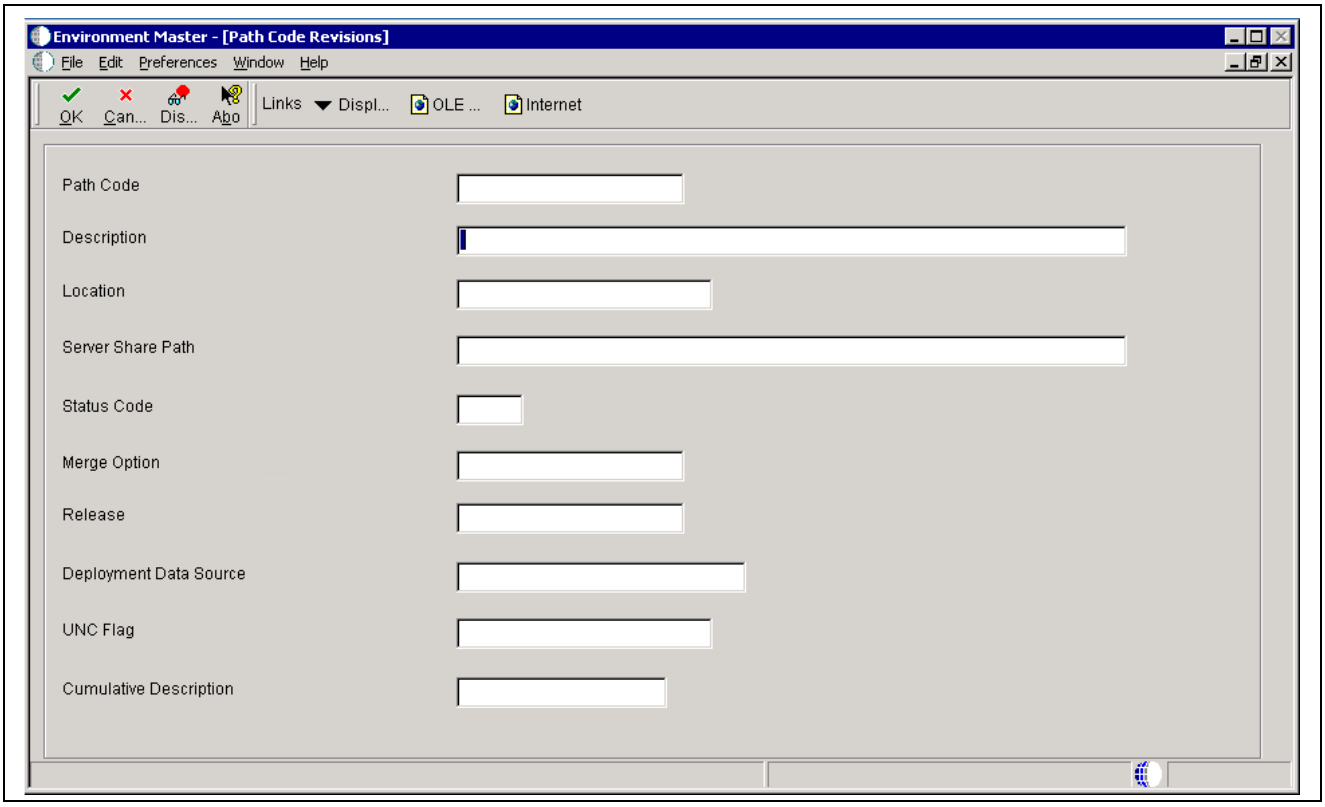
Forms Used to Copy New Environment to a New Path Code

Form Name	FormID	Navigation	Usage
Work With Environments	W0094E	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094). In the Work With Environments form, click Find.	Used to copy a new environment to a new path code.
Copy Environment	W0094B	In the Work With Environments form, highlight the row you want and click Copy Environment.	Used to copy and environment.
Path Code Revisions	W980042D	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094). In the Work With Environments form, click Find. Highlight the new environment and click Path Codes for the Row menu.	Used to copy a new environment to a new path code.

Copying a New Environment to a New Path Code

Copying a new environment to a new path code requires that you know how to use the copy feature to create a new test environment with demo data loading in a new path code.

Access the Path Code Revisions form.



Path Code Revisions form

Path Code Specify a pointer to a set of JD Edwards EnterpriseOne objects which will be used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Description Specify a user defined name or remark.

Location Indicate the name of the machine on the network (server or workstation).

Server Share Path Indicate the shared directory for the path code. The objects that are stored on a file server will be found in this path.

Status Code Determine the status of the software in the development cycle.

Merge Option Denote whether a customer's object will be merged with the JD Edwards EnterpriseOne object This option can be set at the path code level so that all objects checked into that path will carry the same merge option as the path code.

Release Number Specify the release number as defined by the release master.

Deployment Data Source Specify the data source if the primary data source or if the data item in the primary data source cannot be located.

UNC Flag Determine how to create the serve path. Valid options are:

1 or Y Creates the paths using relative paths. Enter a double slash (||) rather than a specific drive followed by a single slash (\).

0 or N Creates the path using the actual drive letter.

Cumulative Description The description that appears in the Z record of a SAR when OMW enters information into that record.

Creating Test Batch Files

This section discusses how to create test batch files.

Creating Test Batch Files

To create test batch files:

1. Using a text editor such as Notepad, remark out all lines in the LOADALL and JDESET files and format a test set in the same format as the production set.
2. Copy the LOADPROD to a LOADTEST.BAT file, and change the appropriate parameters to the ones you created in the JDESET.BAT file.
3. Run the LOADALL.BAT batch application.
4. Add the data sources you need in the current plan's planner environment.
   - For logical data sources, you will probably want Business Data Test, Central Objects Test, JD Edwards EnterpriseOne Local Test, and Control Tables Test.
   - Make sure the deployment server source matches the Central Objects Data source and the LOADALL specifications that you entered.

Updating Server Map Tables with the New Environment

This section discusses how to update server map tables with the new environment

Form Used to Update Server Map Tables

Form Name	FormID	Navigation	Usage
Work With Installation Plans	W98240B	In Solution Explorer, navigate to System Administration Tools, System Installation Tools, Custom Installation Plan (P98240). In the Work With Installation Plans form, select your plan and expand it. Select the environment and click Select. In the Row menu, click Copy Environment. Save the environment you added.	Used to update the server map tables with the new environment.
Work with Locations and Machines	W9654AA	In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Advanced Operations. In the Work with Locations and Machines form, select your location and expandJD Edwards EnterpriseOne Servers. Select the server you want to reuse. Select Generate Server Mapfrom form menu. Select Generate Svr. Map from the Form menu	Used to revise server map tables.

Adding a New Path Code on the Enterprise Server

To add a new path code on the JD Edwards EnterpriseOne Server:

1. Sign on to the JD Edwards EnterpriseOne server with an account that has authority to the existing path codes.
2. Navigate to an existing path code.
3. Copy that directory to the new path code name.

Windows Path Example

This is an example of a Windows path:

D:\JDEdwards\E900\ddp\PS900 (copy PS900 to TS900)

Unix Path Example

This is an example of a Unix path:

/u01/jdedwards/E900/PS900 (copy PS900 to TS900)

iSeries Path Example

You can also map a network drive to the root directory and do the first step from the deployment server.

1. MKDIR TS900 CPY OBJ('/PS900/*') SUBTREE(*ALL) OWNER(*KEEP)
2. Then copy the path code library: ADDLIBLE E900SYS CPYLIB FROMLIB(PS900) TOLIB(TS900) CRTLIB(*YES)

Deleting an Environment

This section discusses how to delete an environment.

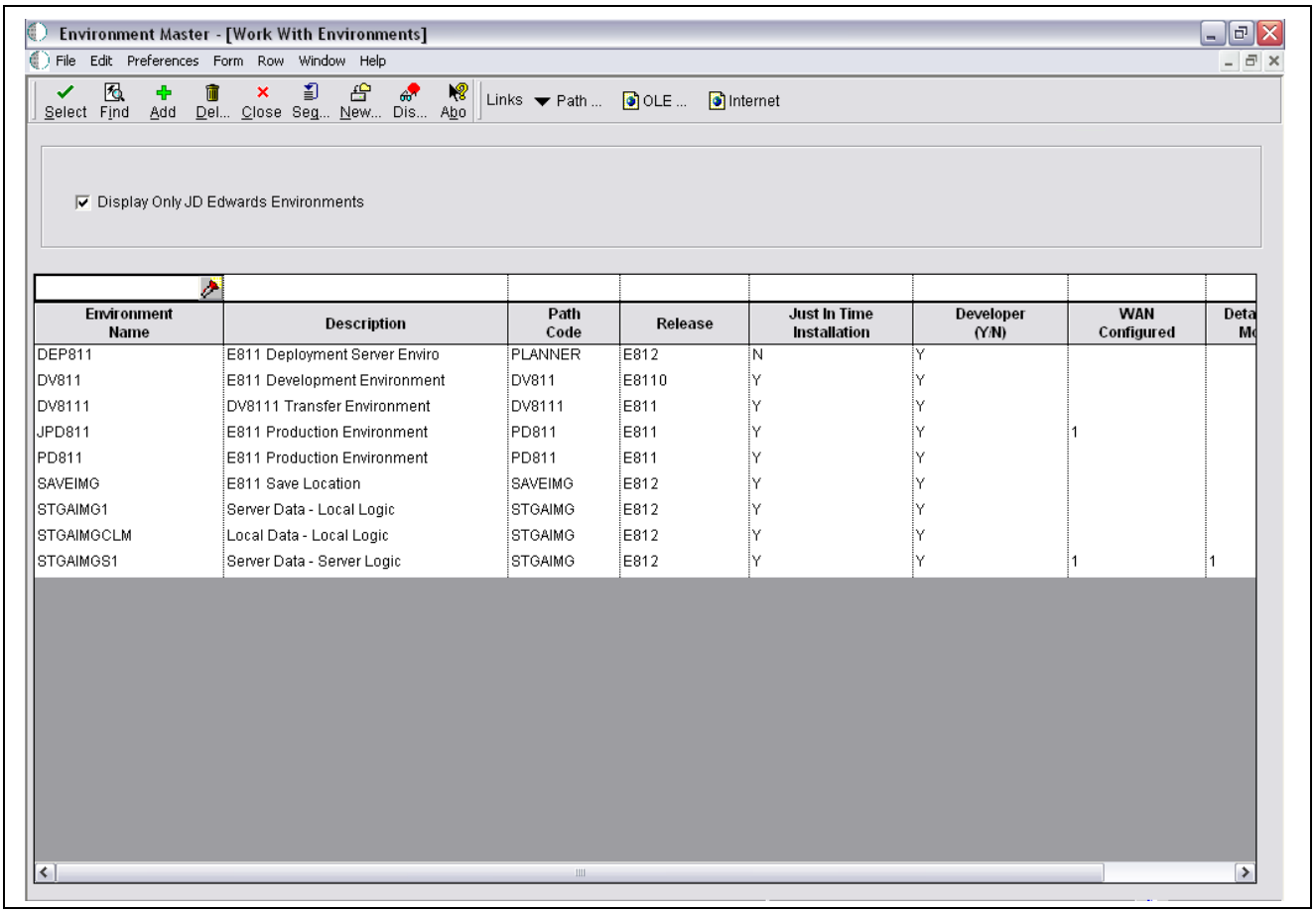
Form Used to Delete an Environment

Form Name	FormID	Navigation	Usage
Work With Environments	W0094E	System Administration Tools, System Administration Tools, Environment Management, Environment Master (P0094). In the Work With Environments form, click Find. Highlight the row you want and click Select. Under File, click Delete. In the Confirm Delete message, click OK.	Used to delete an environment.

Deleting an Environment

When you delete an environment, the environment definition is removed and the Object Configuration Manager records associated with the environment are deleted. You can set a processing option if you want to specify additional Object Configuration Manager mappings to delete when you delete the environment.

Access the Work With Environments form.



Work With Environments form

Environment Name Specifies the name to be associated with a specific list of libraries. The J98INITA initial program uses these library list names to control environments that a user can sign on to. These configurations of library lists are maintained in the Library List Master File table (F0094).

This field represents a valid environment that can be used to run in JD Edwards EnterpriseOne. The environment encompasses both a path code (objects) and a data source (data). When put together, users have a valid workplace within the system.

Description Specifies a user defined name or remark.

Path Code Specifies the path code as a pointer to a set of JD Edwards EnterpriseOne objects, and is used to keep track of sets of objects and their locations within JD Edwards EnterpriseOne.

Release Specifies the release number as defined in the Release Master.

Just In Time Installation Used to turn ON/OFF just-in-time installation for anyone signed onto this environment. Consider turning just-in-time installation OFF before you transfer modified applications into the production path code. Once you have fully tested the applications and are ready for production users to receive the changes, you can turn just-in-time installation back ON.

Developer (Y/N) Future Use.

	A one byte field for the Install group.
WAN Configured Environment	Indicates that this environment is an environment that is configured for the WAN. *1* - WAN Configured Environment *Blank* - Not Configured for WAN
Mobile Environment	Indicates that this environment is a mobile environment.

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*.
softcoding	A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.
source repository	A repository for HTTP adapter and listener service development environment artifacts.
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

S

T

V

W