

JD Edwards EnterpriseOne Tools

System Administration Guide

9.2

Copyright © 2011, 2023, Oracle and/or its affiliates.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	i
<hr/>	
1 Introduction and Understanding of JD Edwards EnterpriseOne Tools System Foundation and Administration	1
Oracle's JD Edwards EnterpriseOne Architecture Overview	1
EnterpriseOne Architecture Implementation	1
Implementation Teams	1
EnterpriseOne Architecture Foundation Overview	3
EnterpriseOne Architecture Advantages	3
EnterpriseOne Architecture Fundamentals	5
2 Understanding Typical Customer Configurations	11
Recommended Configurations	11
Configuration Data	22
3 Understanding Path Codes	25
Understanding Path Codes	25
Setting Up Path Codes	29
4 Understanding and Setting Up Data Sources	31
Data Sources	31
Required Data Source Types	32
Database Structures for EnterpriseOne	35
System Data Source Connections	43
System Table Caching	44
Understanding Data Source Set Up	45
Adding or Modifying a Database or Logical Data Source	46
Defining an Alternate Data Source for EnterpriseOne Search (Release 9.2.1.2)	51
Modifying the Release/Data Source Map Table	52
Understanding Data Source Reports	56

Running the Data Source Master and Data Source Master Compare Reports	56
Running the Verify Object Configuration Mappings Report	61

5 Understanding and Setting Up Environments 67

Understanding Environments	67
Setting Up Environments	69
Adding an Environment	71
Setting Up Color Coded Environments (Release 9.2.4)	73
Copying an Environment	74
Copying a New Environment to a New Path Code	76
Creating Test Batch Files	78
Updating Server Map Tables with the New Environment	79
Deleting an Environment	80
Understanding Copying an Environment to a New Environment	82
Understanding Creating OCM Mappings	83
Understanding UBE Copy Programs	85
Setting Up Database Components	86
Configuring Setup Files	87
Setting Up Security Overrides	90
Using Environment Director to Copy an Environment to a New Environment	91
Using Environment Director in the Express Mode	110
Running Environment Director from Different Environments	118
Using Object Management Workbench to Modify Table Data Classes	120
Using On Track Planning Setup to Modify Table Data Classes	122

6 Understanding and Working with Object Configuration Manager 127

Object Configuration Manager Overview	127
Distributed Architecture	129
Partitioning Application Logic on Servers	129
Understanding Object Configuration Manager	132
Setting Up Object Configuration Manager	134
Setting Up Object Mappings for the Object Librarian Table	135
Changing Mappings for an Object Librarian Table	139
Updating the Object Configuration System Table	142
Creating OCM Records for Business Functions	145
Updating the Oracle Parameters Table	148
Revising the Generic Text Language Status Table	155

Understanding Object Configuration Management Reports	158
Running Job Master Deletion by Days Old Report	158
Running Object Configuration Interactive and Batch Applications	163
Running the Object Configuration Mapping Comparison Report	166
Running the Object Configuration Global Update Report	167
Running the Object Configuration Delete Report	168
Running the Object Configuration Copy Report	169
Running the OCM Category Update/Delete Report	170
7 Understanding Application Communication	173
Middleware	173
JDENet Communication Middleware	173
JDEBase Database Middleware	176
Working with Direct-Connect Processing	177
8 Using the Scheduler Application	179
Understanding the Scheduler Application	179
Working with the Job Scheduler	180
Using Advanced Scheduling Options	185
Reviewing the Job Schedule	189
Working with the Scheduler Server	192
Creating Daylight Savings Rules	196
Running Scheduler Reports	198
9 Glossary	201
activity rule	201
CafeOne	201
charts	201
composite application framework	201
edit mode	201
fast path	201
processing option	202
QBE	202
selection	202
softcoding	202
visual assist	202

vocabulary override	202
Index	203

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Information

For additional information about JD Edwards EnterpriseOne applications, features, content, and training, visit the JD Edwards EnterpriseOne pages on the JD Edwards Resource Library located at:

<http://learnjde.com>

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
<i>Italics</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
Monospace	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step-by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Introduction and Understanding of JD Edwards EnterpriseOne Tools System Foundation and Administration

Oracle's JD Edwards EnterpriseOne Architecture Overview

Oracle's JD Edwards EnterpriseOne Architecture 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.

EnterpriseOne Architecture Implementation

EnterpriseOne standardizes and automates software installation, making many steps transparent to users. Technical setup is preconfigured to meet the requirements of many EnterpriseOne customers. In addition, 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.

Implementation Teams

The 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 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 EnterpriseOne applications, as well as the modeling of all business processes that will be realized through the application set. After 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 EnterpriseOne staff as EnterpriseOne Architect consultants, network architects, custom modification consultants, and so on. In addition, customers provide hardware and network infrastructure support.

Implementing the EnterpriseOne system includes many tasks that are outside the scope of EnterpriseOne software and services. Systems integration (that is, third-party) consultants provide these services to help you align the infrastructure

to optimally support 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 EnterpriseOne software solution. These project members should be aware of the architecture and technical behavior of EnterpriseOne software and of how the various technology components interact with operating systems, database management systems, third-party middleware, and the network.

EnterpriseOne Architecture Foundation Overview

EnterpriseOne architecture is the technical architecture for EnterpriseOne software. EnterpriseOne 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. EnterpriseOne architecture insulates the business solution from the underlying technology. Enterprises can grow and adopt new technologies without rewriting applications.

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.

EnterpriseOne Architecture Advantages

This section discusses the advantages that the EnterpriseOne 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

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. 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.

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 EnterpriseOne applications.

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

Worldwide Business Support

EnterpriseOne provides support for mixed currency and languages. Also, you can run 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 EnterpriseOne. The 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 EnterpriseOne that you can customize without consequences during an upgrade:

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

EnterpriseOne Architecture Fundamentals

This section discusses the fundamentals of the EnterpriseOne architecture, which consists of these items:

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

Environments

An EnterpriseOne environment is a collection of pointers indicating the location of data and 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?

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.

Question	Response
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 EnterpriseOne executes (the directory in which they reside). This location is called a path code, and 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 EnterpriseOne uses at run time.

Shared Object Server

Contains a replicated set of objects that EnterpriseOne Enterprise and HTML 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 EnterpriseOne.

Data Sources

A data source is the specific location of data or distributed processing. 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 for i*.
- 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 EnterpriseOne.

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

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

Oracle DBMS

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

Oracle OEE

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

SQL Server DBMS

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

DB2 for i DBMS

An EnterpriseOne data source for a *DB2 for i* DBMS points to a RDB directory entry and a Library (ODBC data source).

MSDE DBMS

An 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. 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

EnterpriseOne provides three general storage formats; central objects, package objects, and serialized objects to accommodate several functions in 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 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 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 EnterpriseOne server shares a spec package contained in an enterprise RDMS with other EnterpriseOne servers and other web servers. An 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 EnterpriseOne specifications into Java code, which enables you to access EnterpriseOne applications in HTML. The EnterpriseOne forms and applications that are generated are HTML objects. EnterpriseOne stores the objects in the local database and retrieves them at runtime. The serialized objects serve the function of a persistent cache.

End User UDOs

User Defined Objects (UDOs) are personalized objects that you can view, create, and share for your own use, depending on the permissions you have been granted. The following list contains all UDOs:

EnterpriseOne Pages are the first screen end-users see when they log into EnterpriseOne. They are HTML files, and can contain any HTML-enabled functionality, such as interactive process flows, URL links and web pages, company logos, etc. You must be proficient in coding HTML to configure or create EnterpriseOne Pages.

CafeOne is an abbreviation for the Composite Application Framework, which is a user interface framework that enables EnterpriseOne applications to integrate with multiple third-party contents and applications, as well as other EnterpriseOne applications.

Queries enable you to select fields and QBE columns from a form and add conditions to make the search criteria more specific. The query feature is enabled on find browse, search/select, and power browse forms that have a Find button. Additionally, you can create queries in Data Browser for records in tables and business views.

Watchlist is a collection of items that match user-defined criteria and contain information to which users have selected to be alerted.

Grid Formats are a selection of the columns you choose to display, and the sequence in which they are displayed. Grid formats enable you to customize how information is displayed in your grid.

One View Reports (OVRs) consist of A One View Report consists of:

- A Business Intelligence (BI) Publisher data model, which is a <report name>.xdmz file on the BI Publisher server.
- A BI Publisher report, which is a <report name>.xdoz file on the BI Publisher server.

-
- The report definition for the One View Report, which is the metadata for the report and resides in EnterpriseOne.

Object Deployment

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

- Initial installation, for workstations and servers.
- Workstation installation, for workstations.
- Update packages for individual objects.

Initial Installation

The installation process is based on a centralized deployment server model. The Deployment Server Installation program copies EnterpriseOne installation software from the extracted downloaded file location to the deployment server. From the deployment server, you redistribute the software to the enterprise servers and workstations.

Workstation Installation

The Workstation Installation program 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 EnterpriseOne can read at runtime.

Update Packages for Individual Objects

Objects can also be deployed by using update packages for individual objects.

See *JD Edwards EnterpriseOne Applications Release 9.2 Upgrade Guide (for your database and platform)*.

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

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: HTML server, 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.

2 Understanding Typical Customer Configurations

Recommended Configurations

EnterpriseOne can be configured in many ways. Examples and recommendations for the setup follow.

You should follow 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 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 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 EnterpriseOne share common data sources. For example, for EnterpriseOne 8.11, these data sources are common to all environments:

- System - 910
- Object Librarian - 910
- Data Dictionary - 910

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 (PD910)

This table explains the data sources for the production environment.

Data Source	Environment
Path Code	PD910
Central Objects data source	Central Objects - PD910
Versions Tables data source	Versions - PD910
Business Data data source	Business Data - PROD
Next Numbers data source	Control Tables - Prod
Tasks/UDCs data source	Control Tables - Prod

Prototype Environment (PY910)

This table defines the data sources for the prototype environment.

Data Source	Environment
Path Code	PY910
Central Objects data source	Central Objects - PY910
Versions Tables data source	Versions - PY910
Business Data data source	Business Data - CRP
Next Numbers data source	Control Tables - CRP
Tasks/UDCs data source	Control Tables - CRP

Development Environment (DV910)

This table defines the data sources for the development environment.

Data Source	Environment
Path Code	DV910
Central Objects data source	Central Objects - DV910
Versions Tables data source	Versions - DV910
Business Data data source	Business Data - TEST
Next Numbers data source	Control Tables - Test
Tasks/UDCs data source	Control Tables - Test

Pristine Environment (PS910)

This table defines the data sources for the pristine environment.

Data Source	Environment
Path Code	PS910
Central Objects data source	Central Objects - PS910
Versions Tables data source	Versions - PS910
Business Data data source	Business Data - PS910
Next Numbers data source	Business Data - PS910
Tasks/UDCs data source	System Local - PS910

Planner Environment (PSFTPLAN)

This table defines the data sources for the deployment environment.

Data Source	Environment
Path Code	PLANNER

Data Source	Environment
Central Objects data source	Not applicable
Planner Tables data source	Planner - 910
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

Deployment Environment (DEP910)

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:

xxxJPD910 indicates a Java application server for the PD910 environment at location xxx.

The path codes and data sources for remote environments are identical to the base environments.

PD910 Environment

This table defines the data sources for a remote production environment.

Data Source	Environment
Path Code	PD910
Purpose	PD910 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.

PD910 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 - PD910	Maps to the central object tables, including F98950 - User Overrides.
Control Tables - Prod	Maps to the next number tables.
Data Dictionary - 910	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 - 910	Maps to the object librarian tables and the object librarian media object text in these data structures: <ul style="list-style-type: none"> 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 - PD910	Maps to the user defined code tables.
System - 910	Maps to the system tables.
Versions - PD910	Maps to the version tables, including: <ul style="list-style-type: none"> F983051 - Versions List

Mapping	Explanation
	<ul style="list-style-type: none"> F98306 - Processing Option Text

PY910 Environment

This table defines the data sources for a remote prototype environment.

Data Source	Environment
Path Code	PY910
Purpose	<p>PY910, 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.</p> <p>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 PD910, which represents the production data.</p>

PY910 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 - PY910	Maps to the central object tables, including F98950 - User Overrides.
Control Tables - CRP	Maps to the next number tables.
Data Dictionary - 910	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 - 910	<p>Maps to the object librarian tables and the object librarian media object text in these data structures:</p> <ul style="list-style-type: none"> 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 - PY910	Maps to the user defined code tables.

Mapping	Explanation
System - 910	Maps to the system tables.
Versions - PY910	Maps to the version tables including: <ul style="list-style-type: none"> F983051 - Versions List F98306 - Processing Option Text

DV910 Environment

This table defines the data sources for a remote development environment.

Data Source	Environment
Path Code	DV910
Purpose	<p>DV910, the development environment, is the testing environment for development objects. This environment shares the test data that TS910 uses. Developers log in to this environment to modify objects and test them before transferring the changed objects to the PD910 path code.</p> <p>Once you have transferred objects into PD910, 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.</p>

DV910 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 - DV910	Maps to the central object tables, including the F98950 - User Overrides table.
Control Tables - Production	Maps to the next number tables.
Data Dictionary - 910	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.
Object Librarian - 910	Maps to the object librarian tables and to the object librarian media object text in these data structures: <ul style="list-style-type: none"> GT9860A - Object Librarian Generic Text Structure GT9862A - Business Function Notes

Mapping	Explanation
	<ul style="list-style-type: none"> GT98DSA - Data Structure Notes - Structure and Item GT98DSB - Data Structure Notes - Structure and Item GT98TMPL - Media Object Templates
Local - DV910	Maps to the user defined code tables.
System - 910	Maps to the system tables.
Versions - DV910	Maps to the version tables, including; <ul style="list-style-type: none"> F983051 - Versions List F98306 - Processing Option Text

PS910 Environment

This table illustrates the data sources for a remote test environment.

Data Source	Environment
Path Code	PS910
Purpose	<p>Use the PS910 environment to test pristine (unaltered) objects with EnterpriseOne demonstration data. Also use this environment for training classes. You must have this environment to compare modified objects to pristine objects.</p> <p>When you encounter a software problem that 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 EnterpriseOne demonstration data shipped with the software.</p>

PS910 Object Mappings

This table illustrates the object mappings for a test environment.

Mapping	Explanation
Business Data - PS910	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.
Data Dictionary - 910	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 - 910	Maps to the object librarian tables and to the object librarian media object text in these data structures:

Mapping	Explanation
	<ul style="list-style-type: none"> 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 - PS910	Maps to the user defined code tables.
System - 910	Maps to the system tables.
Versions - PS910	Maps to the version tables, including: <ul style="list-style-type: none"> 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 - PS910	The owner is PS910DTA. The library is PS910DTA. The pristine data shipped with the software.
Business Data - PROD	The owner is PRODDTA. The library is PRODDTA. The production business data.
Business Data - TEST	The owner is TESTDTA.

Data Source Name	Owner and Purpose
	<p>The library is TESTDTA.</p> <p>The test data entered during CRP or converted from non-EnterpriseOne systems.</p>
Central Objects - PY910	<p>The owner is PY910.</p> <p>The library is COPY910.</p> <p>The central objects data source associated with the PY910 path code. After you create and test the modifications in this path code, transfer them to the PD910 path code.</p>
Central Objects - DV910	<p>The owner is DV910.</p> <p>The library is CODV910.</p> <p>The central objects data source associated with the DV910 path code. After you create and test the modifications in this path code, transfer them to the PD910 path code.</p>
Central Objects - PS910	<p>The owner is PS910.</p> <p>The library is COPS910.</p> <p>The central objects data source associated with the PS910 path code.</p>
Central Objects - PD910	<p>The owner is PD910.</p> <p>The library is COPD910.</p> <p>The central objects data source associated with the PD910 path code. Transfer objects into this data source after you have tested them in path code DV910.</p>
Control Tables - CRP	<p>The owner is CRPCTL.</p> <p>The library is CRPCTL.</p> <p>The control tables used in the CRP environment.</p>
Control Tables - Prod	<p>The owner is PRODCTL.</p> <p>The library is PRODCTL.</p> <p>The control tables used in the production environment.</p>
Control Tables - PS910	<p>The owner is PS910CTL.</p> <p>The library is PS910CTL.</p> <p>The control tables used in the PS910 path code.</p>
Control Tables - Test	<p>The owner is TESTCTL.</p> <p>The library is TESTCTL.</p> <p>The control tables used in the test environment.</p>

Data Source Name	Owner and Purpose
Data Dictionary - 910	<p>The owner is DD910.</p> <p>The library is DD910.</p> <p>The single data dictionary that all environments use.</p>
LOCAL	<p>No owner exists.</p> <p>The data source that defines the local machine to EnterpriseOne. Use it to override reports to the workstation.</p>
machine name	<p>No owner exists.</p> <p>A value that defines the logic host to EnterpriseOne.</p>
Machine Name - Server Map	<p>The owner is SVM910.</p> <p>The server map for the logic server.</p>
Object Librarian - 910	<p>The owner is OL910.</p> <p>The Object Librarian, which is release specific.</p>
Control Tables Local	<p>No owner exists.</p> <p>The data found in the PSCTL910 MSDE. This data is used only in the planner.</p>
System Local	<p>No owner exists.</p> <p>The data found in the PSSY910 MSDE. This data is used only in the planner.</p>
Data Dictionary Local	<p>No owner exists.</p> <p>The data found in the PSDD910 MSDE. This data is used only in the planner.</p>
Object Librarian Local	<p>No owner exists.</p> <p>The data found in the PSOL910 MSDE. This data is used only in the planner.</p>
Versions Local	<p>No owner exists.</p> <p>The data found in the PSVL910 MSDE. This data is used only in the planner.</p>
Local - PY910	<p>No owner exists.</p> <p>The source that contains the User Defined Codes (UDCs): Control Tables - CRP.</p>
Local - DV910	<p>No owner exists.</p> <p>The source that contains the User Defined Codes (UDCs): Control Tables - Test.</p>

Data Source Name	Owner and Purpose
Local - PS910	<p>No owner exists.</p> <p>The source that contains the User Defined Codes (UDCs): Business Data - PS910.</p>
Local - PD910	<p>No owner exists.</p> <p>The source that contains User Defined Codes (UDCs): Control Tables - Production.</p>
System - 910	<p>The owner is SY910.</p> <p>The library is SY910.</p> <p>The single set of system tables that all environments use.</p>
Versions - PY910	<p>The owner is PY910.</p> <p>The library is COPY910.</p> <p>The source that contains the versions list and processing option text tables for the PY910 environment.</p>
Versions - DV910	<p>The owner is DV910.</p> <p>The library is CODV910.</p> <p>The source that contains the versions list and processing option text tables for the DV910 environment.</p>
Versions - PS910	<p>The owner is PS910.</p> <p>The library is PS910.</p> <p>The source that contains the versions list and processing option text tables for the PS910 environment.</p>
Versions - PD910	<p>The owner is PD910.</p> <p>The owner is COPD910.</p> <p>The source that contains the versions list and processing option text tables for the PD910 environment.</p>

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:

Requirement	Explanation
	<ul style="list-style-type: none"> • Conference Room Pilot (CRP) business data • Conference Room Pilot (CRP) control tables • Data Dictionary (all environments share) • EnterpriseOne pristine data • Production business data • Production control tables (used by the PD910 environment) • Object Librarian (all environments share) • System (technical data all environments share) • Test business data • Test control tables (used by TS910 and DV910 environments) • Versions
Central Objects (Path Codes)	<p>You should have these sets of central objects or path codes, which are release-specific:</p> <ul style="list-style-type: none"> • PY910 central objects • DV910 central objects • PS910 central objects • PD910 central objects
Environment	<p>The installation process defines the environments, which are as follows:</p> <p>PY910</p> <p>CRP objects (possibly testing a package you have not released to production users), with data mapped to CRP. The path code is PY910 with table objects mapped to CRP data.</p> <p>DV910</p> <p>Development objects with test data. The path code is DV910, with table objects mapped to test data.</p> <p>PS910</p> <p>Pristine objects with EnterpriseOne pristine demo data. The path code is PS910, with objects mapped to EnterpriseOne pristine data.</p> <p>PD910</p> <p>Production objects with production data. The path code is PD910, with table objects mapped to production data.</p> <p>TS910</p> <p>CRP objects with test data. The path code is PY910, with table objects mapped to test data.</p>
	<p>If you are not planning any development projects, you need only two path codes: PY910 and PD910. You should create a development path code if you plan to do extensive software modification.</p> <p>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):</p> <p>PY910</p>

Requirement	Explanation
	<p>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.</p>
	<p>DV910</p> <p>Use this path code for normal development. Upon successful testing, transfer the objects to the PY910 path code, using Object Transfer, and distribute to the users through a package build and a workstation installation.</p> <p>PS910</p> <p>This is the set of pristine objects shipped from EnterpriseOne. You should not make changes to this path code other than paper fixes from EnterpriseOne. This path code is used to compare 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.</p> <p>PD910</p> <p>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 PY910, transfer them to PD910 and then deploy the changes to the enterprise servers and workstations.</p>
<p>Packages</p>	<p>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).</p> <p>At 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.</p> <p>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:</p> <ul style="list-style-type: none"> • PD910FA (Standard Production Full A) • PD910FB (Standard Production Full B) • PD910PA (Standard Production Partial A) • PD910PB (Standard Production Partial B)

3 Understanding Path Codes

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 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.

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

EnterpriseOne uses path codes at runtime in these ways:

- To validate available environments.

When you log on to 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 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.

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 EnterpriseOne. For every set of objects in the configuration, EnterpriseOne requires a path code definition in the Object Path Master File table (F00942).

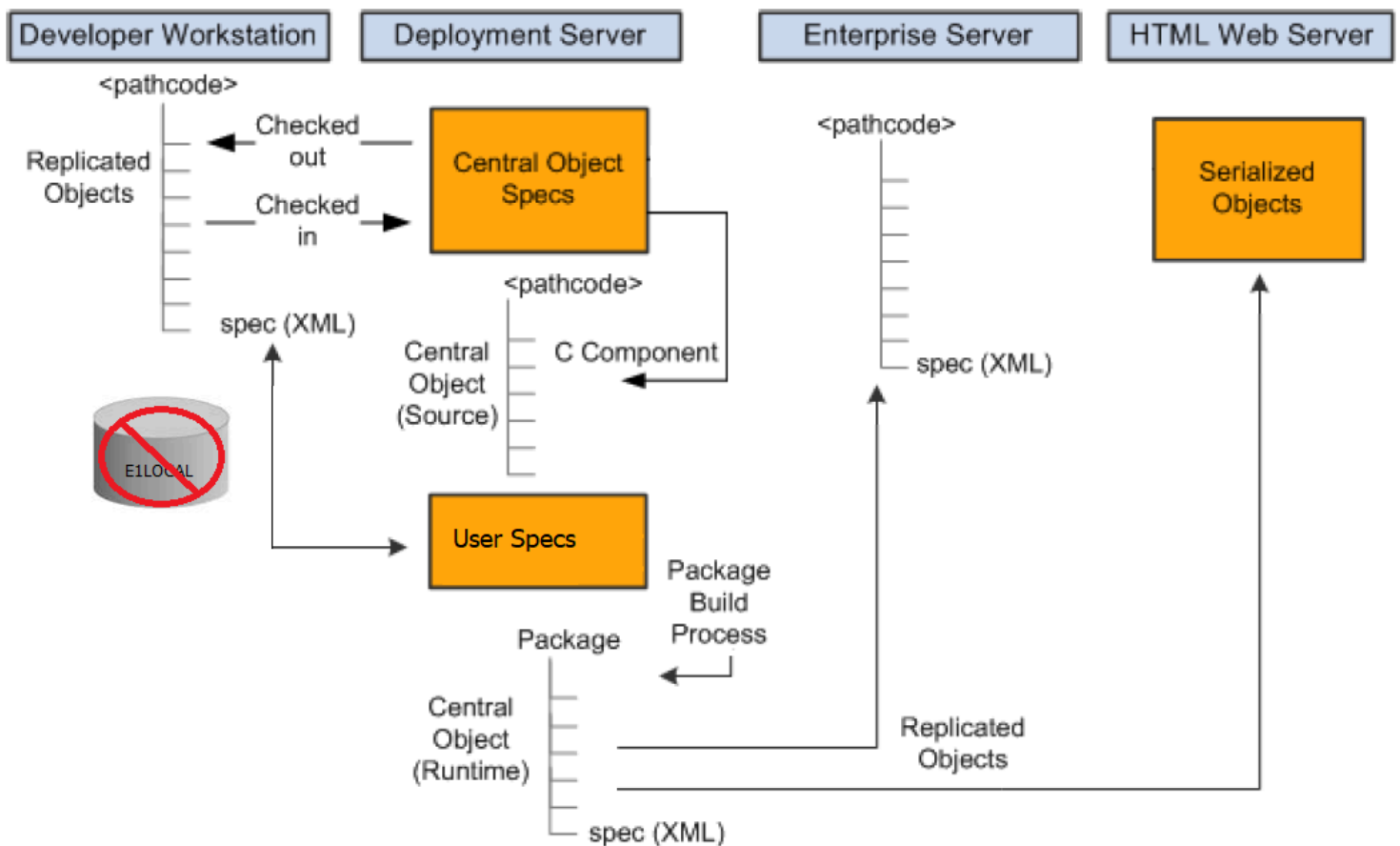
EnterpriseOne recommends a separate path code definition for each of the these sets of objects:

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

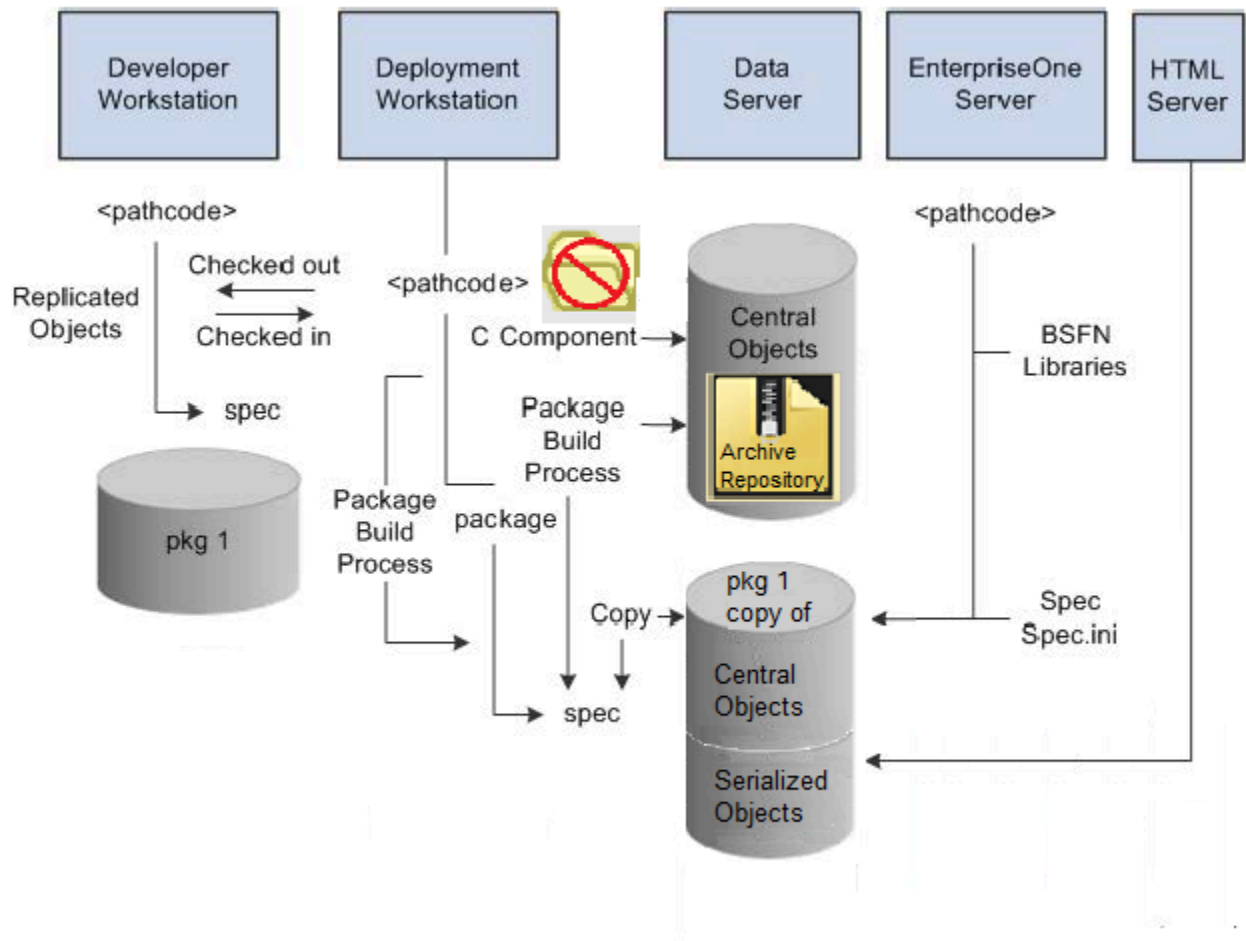
Path Codes and Object Storage

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:



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



The different deployments by releases are:

- Deployment in 9.2.0:
 - Deployment Server Files “source of truth” exist in path code (include/source) folder
 - Files copy with OMW actions (Check-out/in and Transfer)
 - ESU application Package Build uses files from deployment server path code to copy to PACKAGE build folders.
- Deployment in 9.2.1: Files to the Database
 - With package build, files are placed in archives and stored in repository as "source of truth."
 - Files are no longer stored on the deployment server path code folder as "source of truth."
 - Files still exist on the deployment server and still copied with OMW and ESU application.
 - Must build a full package to populate archive repository.

- Deployment in 9.2.3: 64-bit available
 - 64-bit conversion possible (source/include and source64/include64)
 - Package Build - bitness defined by foundation.
 - Code converts if 64-bit path code and build is the first time.
 - 64-bit flag on path code is not set
 - NER generated for bitness of foundation
 - NER does not update source/include (32/64) in database
 - 32-bit NER source/include "could" remain in repository
 - NER spec not needed to be converted to 64-bit - stored as spec only in archive
 - 64-bit development client still develops in 32-bit source
 - OMW check-in and ESU converts 32-bit to 64-bit and stores in archive repository
 - NER stopped storing all include/source in the database repository.
 - One-Click stopped delivering source files in the deployment server for any path code including planner.
 - Must build a full package to create archives and extract from database to compile.
- Deployment in 9.2.4: 64-bit Deployment Server Supported
 - SMT9.2.4.0X64 Special Migration Tools required.
 - 64-bit planner available along with ESU changes.
 - ESUs (and Change Assistant) - 64-bit does not apply (copy) source files to deployment server.
 - If 32-bit applied, source files get copied to path code directory on deployment server.
 - Package Build - updates NER BSFN source only on deployment.
 - If delta detected between repository and files in path code folder.
 - Must build a full package to create archives and maintain 32-bit to 64-bit conversion.
- Deployment in 9.2.4.3: Package Build – No longer copies files to deployment server path codes
 - Package build stopped copying (C or NER) source32/64 or include32/64 from the package directory to the in path code source32/64 and include32/64 directories.
 - User Spec Tables created in central objects data source when Tools Rollup ESU and ASI applied to path code.

- Deployment in 9.2.5: Remove E1Local Database (E1LOCAL DB)
 - No local databases (Pkg Spec or Local) for clients.
Must build a full package since E1Local DB is no longer built into the package.
 - Client Only Packages removed – must include server in build.
 - Web Package Define, Build and Deploy Options available.
New service available on Deployment Server to build client portion of packages.
 - Package Build - No longer updates bin32/64 or lib32/64 in path code or creates E1Local DB.
Package Build creates Serialized Object Tables (F989999 and F989998) for each new full package.
 - OMW no longer copies C BSFN files to deployment server, sends files only to repository tables.
 - OMW provides web transfer option for spec based objects. Files are no longer copied on deployment server between path codes for project status changes.
 - Files in deployment server path code folder are obsolete.

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.

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 Codes

During installation, one path code (PD910) was created for you. The information from this path code can be used to create another one, such as DV910.

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 4.

4 Understanding and Setting Up Data Sources

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 \E910\data, or an *IBM i* 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. 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 EnterpriseOne configuration. Each database and machine in the enterprise must be defined as a data source for 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.

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

See *JD Edwards EnterpriseOne Applications Release 9.2 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 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.

Updating TLS Port for Database Servers (Release 9.2.7.3)

With Tools Release 9.2.7.3, if you have enabled Transport Layer Security (TLS) for Oracle, SQL Server, DB2 for IBMi, or IBM DB2 UDB database, you must specify the TLS port of your database server in the Data Source Master.

To update TLS port:

1. In JD Edwards EnterpriseOne, navigate to EnterpriseOne Menus, EnterpriseOne Life Cycle Tools, System Administration Tools, System Administration Tools, Data Source Management, Database Data Sources.
2. Open Database Data Sources.
3. Open System-920.
4. Click Find to list all the Data Source Names.
5. Select the Data Source where you want to use TLS port.
6. In the left pane, under Form click Port.
7. Update the TLS port in the Database Server Port Number.

Note: The port value varies depending on your database server. It is initially set during installation of EnterpriseOne. You must update this value to use specific TLS port of the database server. Typically, Oracle and IBM DB2 UDB has different ports for both secured and unsecured connections.

Required Data Source Types

You must set up a minimum number of data sources for 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 - 910.

System

This data source consists of the technical tables you use to run all 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
- 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 PD910) and development (such as DV910) 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 - 910, Data Dictionary - B7334, or Data Dictionary - B732.

Local

This data source defines the 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 - PS910.

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 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 - PD910 or Central Objects - DV910.

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 - PD910. It stores versions and processing option information. It includes these tables:

- Versions List (F983051)
- Processing Option Text (F98306)

Alternate Data Source (Release 9.2.1.2)

EnterpriseOne Search execution can be performed on an alternate data source. An alternate data source is a source other than the primary data source that is used for EnterpriseOne operations. If you have a data source with a similar data schema, you can define the search execution to access it instead.

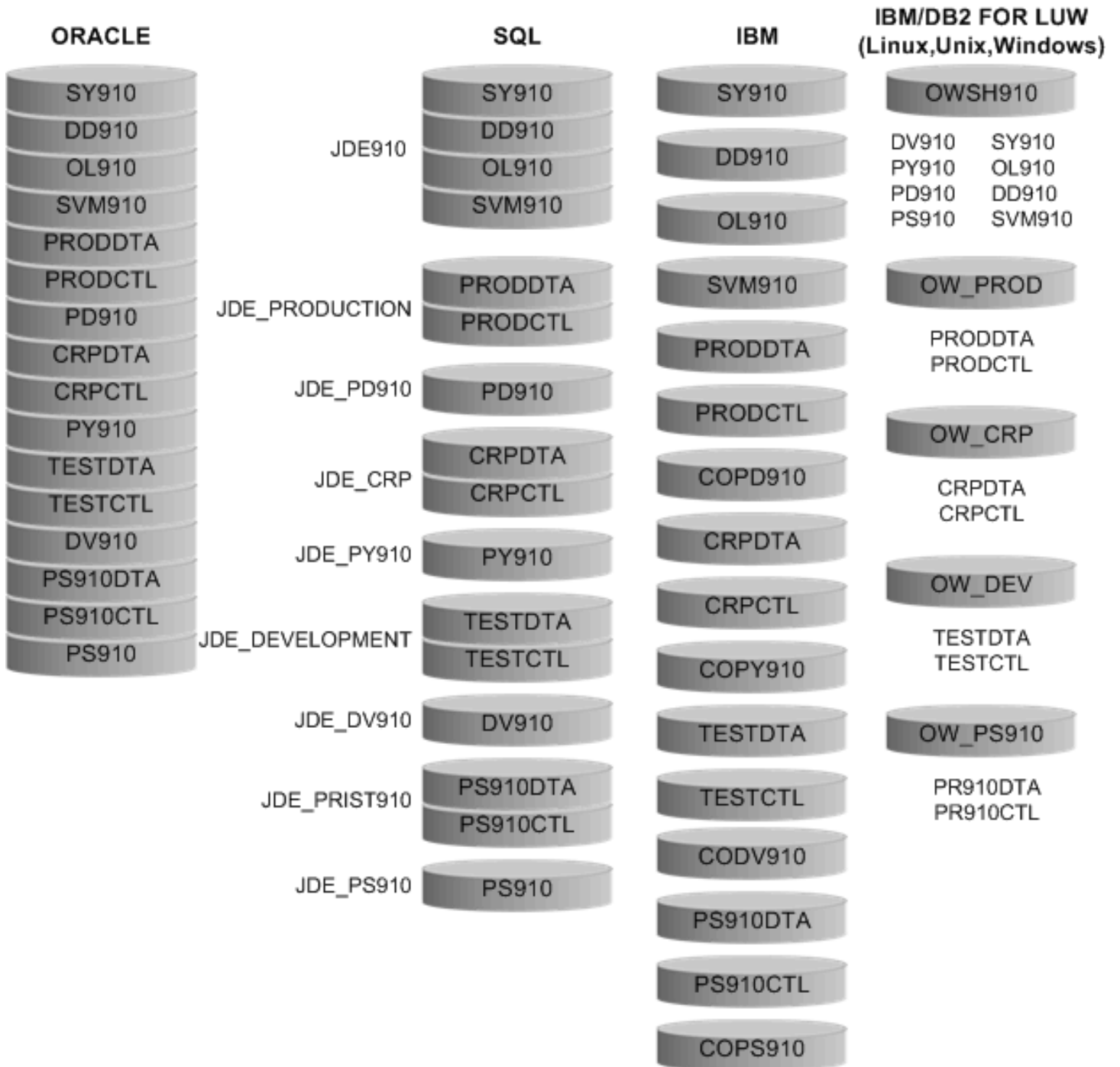
The alternate data source must contain the same data structure, including column names and data types, as the primary data source. On the search result window, the system displays the data retrieved directly from alternate data sources.

See *Defining an Alternate Data Source for EnterpriseOne Search (Release 9.2.1.2)* for more information.

Database Structures for EnterpriseOne

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

This diagram illustrates owners and databases for four different platforms:



Oracle

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.

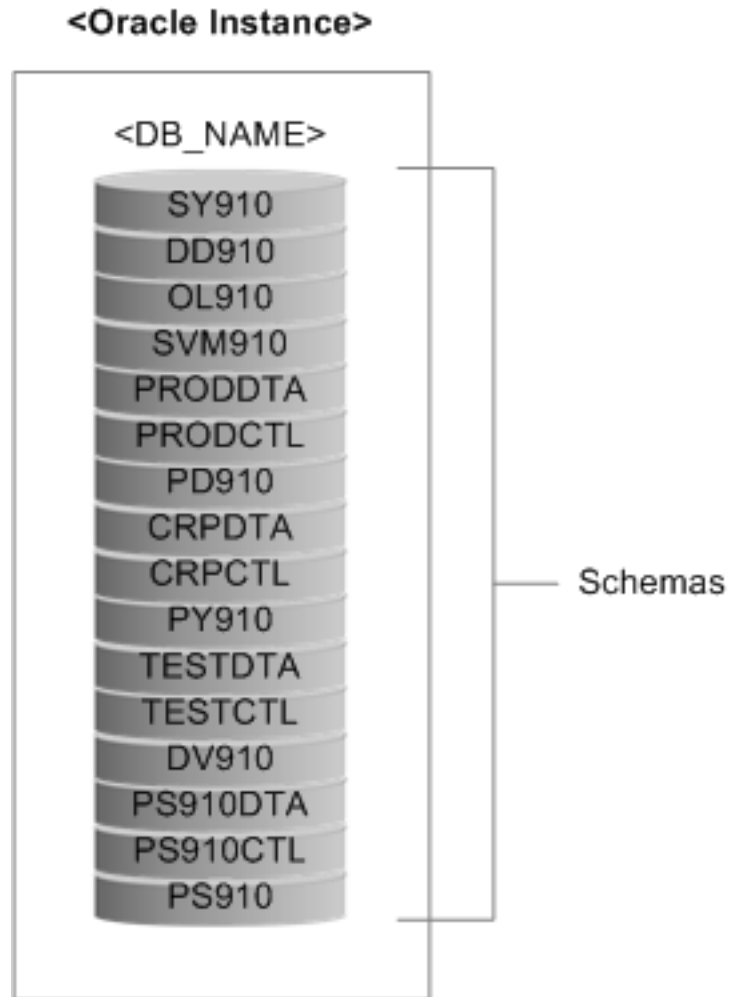
User Schema

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.

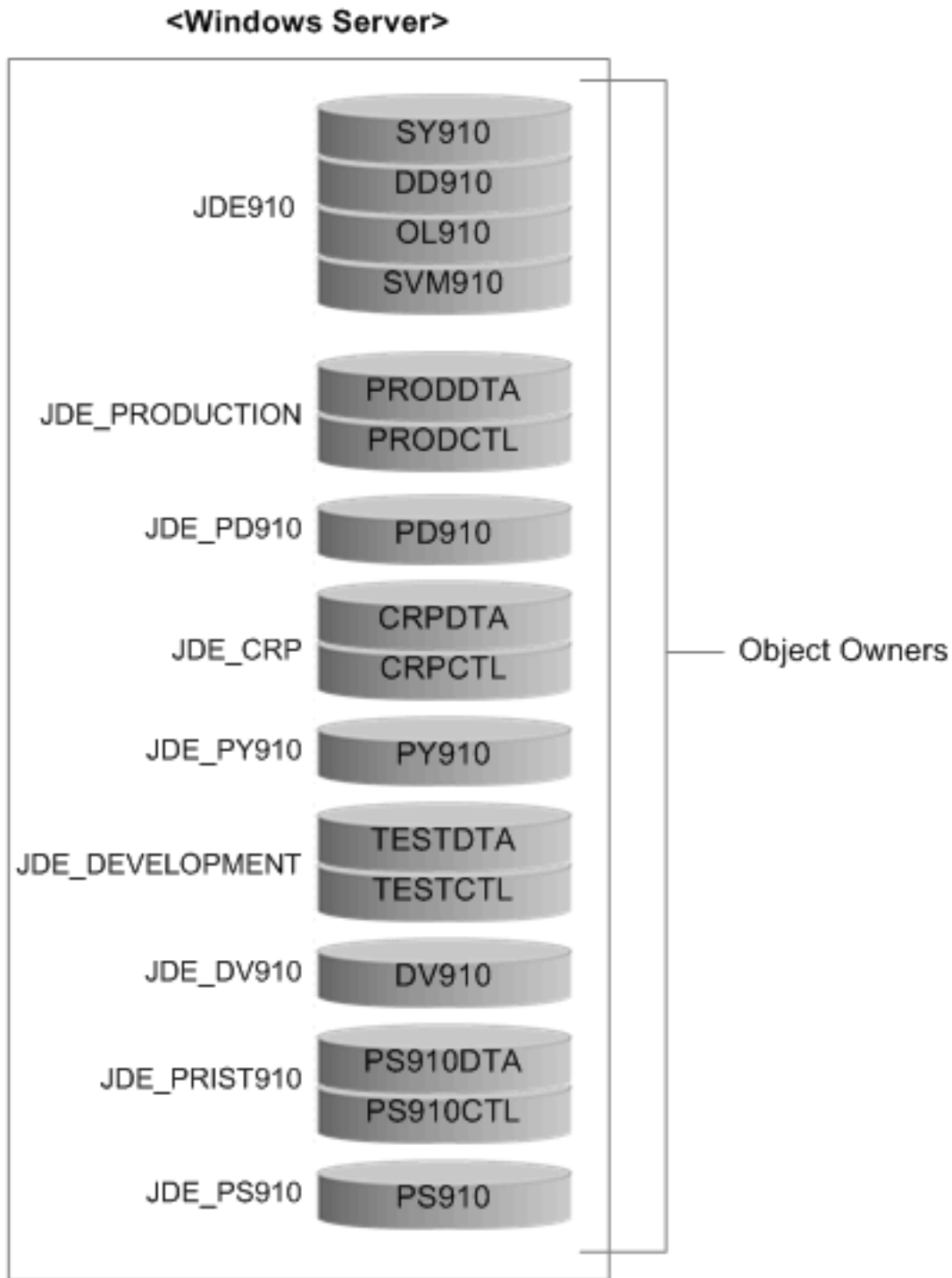
This diagram illustrates the Oracle structure with EnterpriseOne:



SQL Server

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 EnterpriseOne during an installation.

This diagram illustrates the SQL structure with EnterpriseOne:



DB2 for i Server

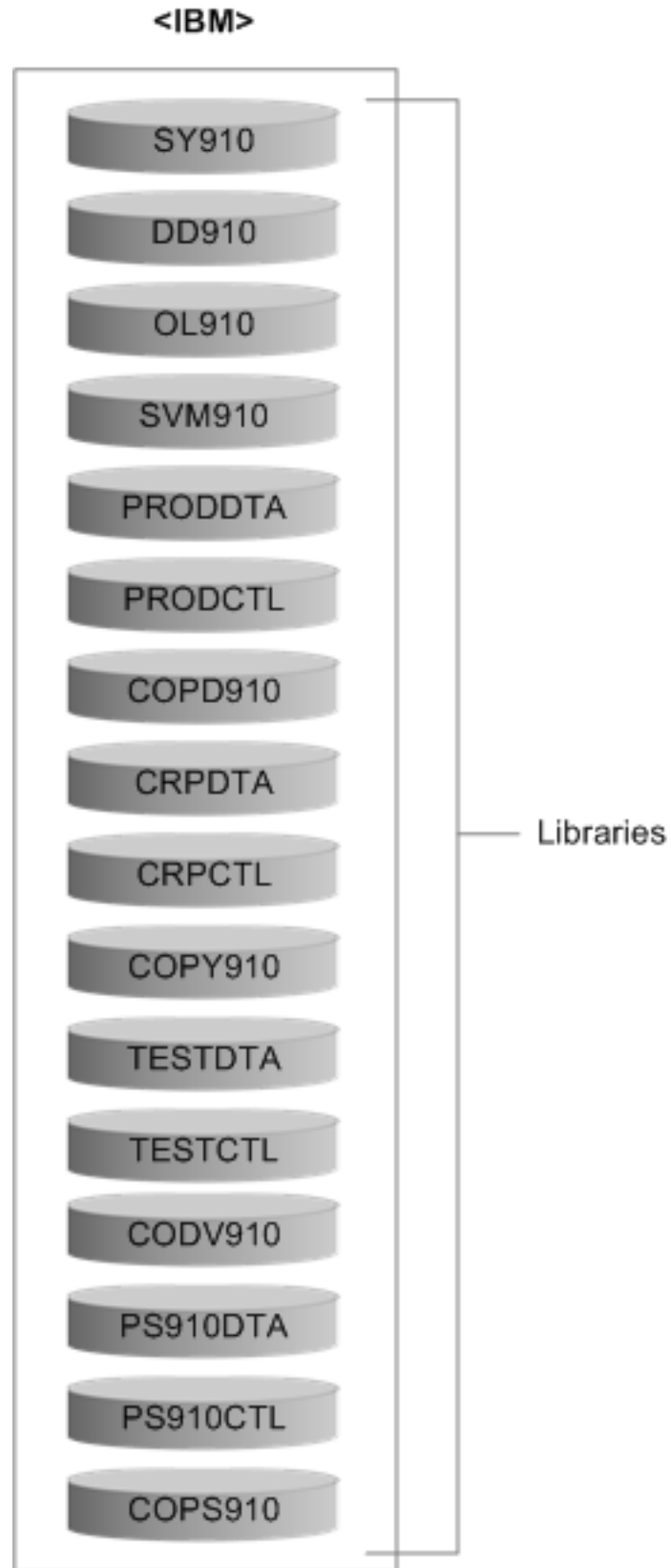
DB2 for i is the relational database manager that is fully integrated and provides numerous functions and features such as triggers, stored procedures, and dynamic bitmapped indexing that serve a wide variety of application types.

These applications range from traditional host-based applications to client/server solutions to business intelligence applications.

In the *IBM i* 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.

DB2 for i installations store all tables in their respective data sources in a single database.

This diagram illustrates the *DB2 for i* structure with EnterpriseOne:



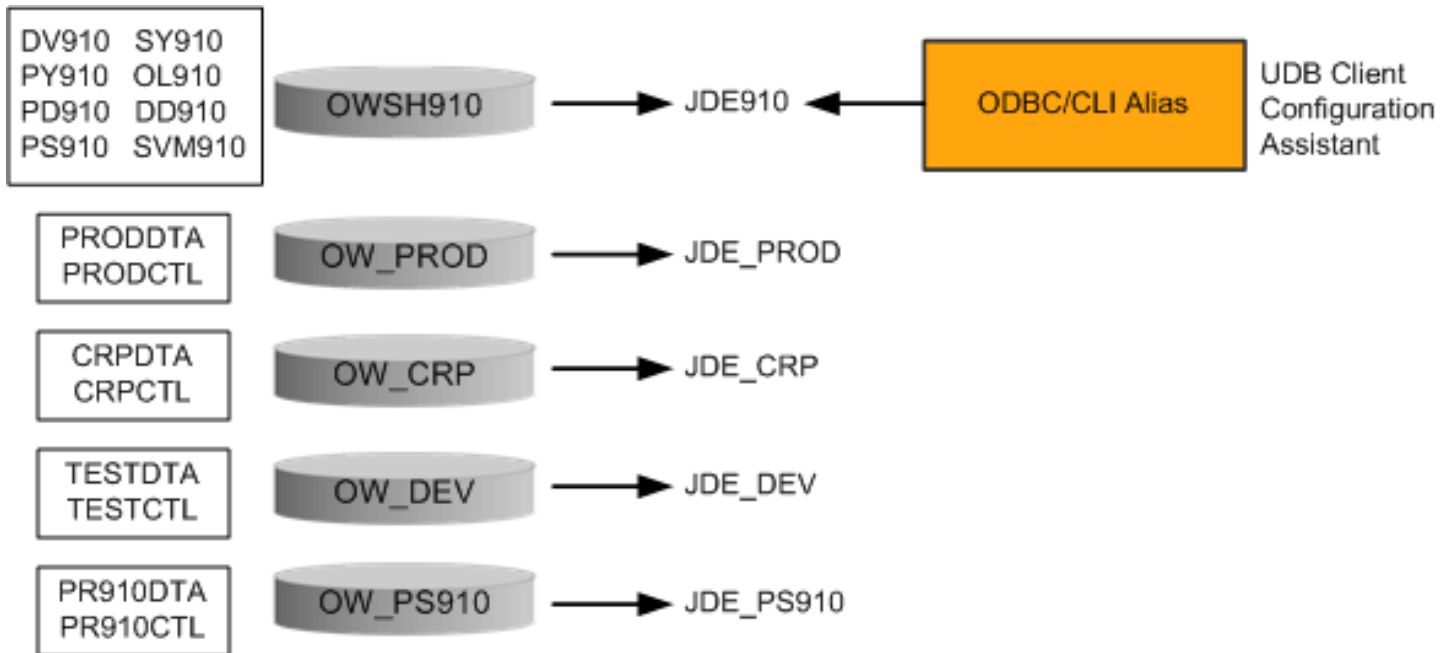
DB2 for Linux, UNIX, and Windows

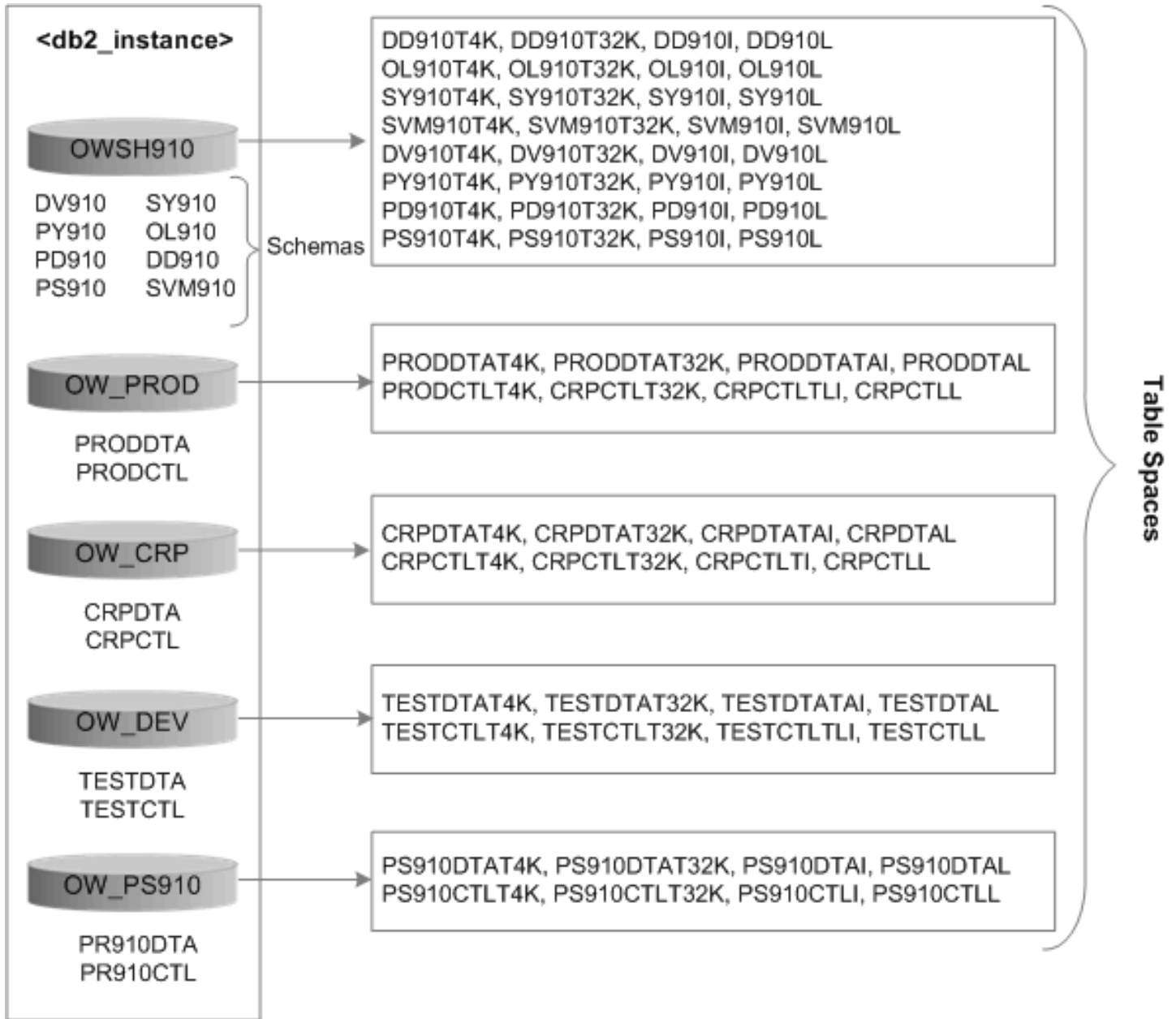
Every data element in a database is stored in a column of a table, and each column is defined to have a data type. The data type places limits on the types of values you can put into the column and the operations you can perform on them. *IBM i* includes a set of built-in data types with defined characteristics 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 for Linux, UNIX, and Windows database 8.1.4* structure with EnterpriseOne:





System Data Source Connections

When 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=DEMO910A
```

```
Default PathCode=DEMO
```

```
Base Datasource=System 910
```

```
Database=System 910
```

```
.  
.  
.
```

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.

Note:

- Major Technical Tables in the JD EdProduct Title/BookTitle as a Variablewards *JD Edwards EnterpriseOne Applications Installation Guide for UNIX with Oracle* .

System Table Caching

When a user first 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 EnterpriseOne session.

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 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 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 PS910 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 *IBM i* 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 E910SYS. 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

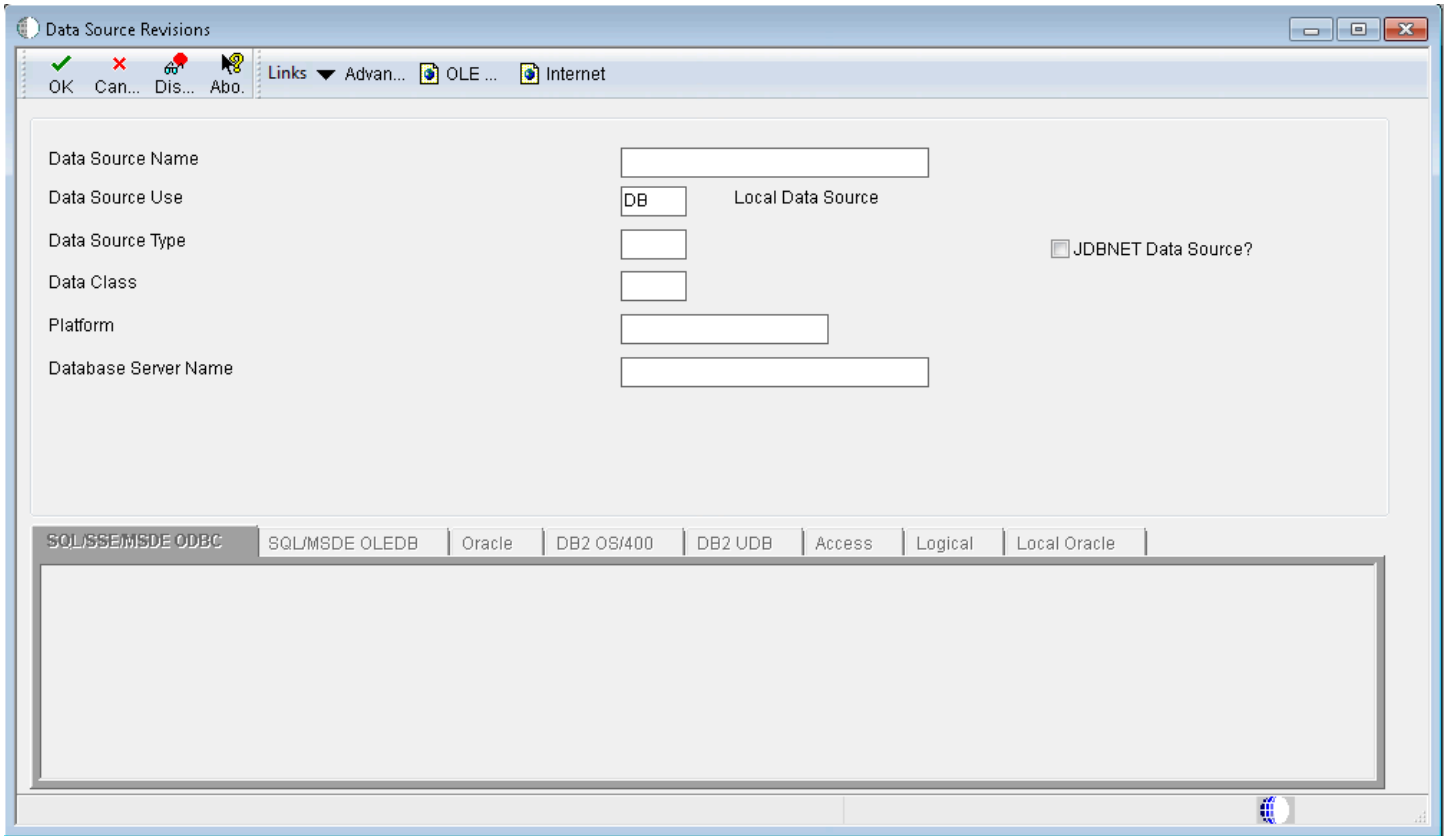
Before you complete the tasks in this section, you should plan for the data sources.

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 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.

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.

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 SQL OLEDB 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.

DB2 for i

Use these options to set up parameters for a *DB2 for i* 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 for Linux, UNIX, and Windows

Use these options to set up parameters for a *DB2 for Linux, UNIX, and Windows database* 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.

Local Oracle

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

Tablespace

Specify the Tablespace.

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 *DB2 for i* data sources.

Defining an Alternate Data Source for EnterpriseOne Search (Release 9.2.1.2)

EnterpriseOne Search can override the primary data source and leverage an alternate data source for search execution. The Alternate Data Sources application (P98611X) enables you to define a data source that is an alternate to a primary data source.

You can designate an alternate data source for a database data source (DB) and not for a server data source (SVR). Each primary data source can only have one alternate data source; however, an alternate data source can have multiple primary data sources pointing to it.

Note: You are responsible for the processing to keep the alternate data source in synchronization with the actual data and determine the frequency of synchronization so that the results are accurate.

To define an alternate data source:

1. In JD Edwards EnterpriseOne, navigate to EnterpriseOne Menu, EnterpriseOne Life Cycle Tools, System Administration Tools, System Administration Tools, Data Source Management, Database Data Sources.
2. In the Machine Search & Select form (W986115E), highlight the machine name and click Select.
3. In the Work With Data Sources form (W986115A), click Find.
4. Select the row for which you want to define an alternate data source.
5. Select Alternate Sources from the Form menu. Alternatively, you can select Alternate Source from the Row exit.
6. In the Alternate Data Sources form (P98611X), specify the name of the Primary Data Source, and then specify the name that identifies the Alternate Data Source. You can also use the Visual Assist to search and select the database data sources.

When you use the Row exit to open the Alternate Data Sources form, the Primary Data Source field displays the data source name and is disabled.

7. Click OK to save and define the alternate data source.

The Search Data Sources table (F98611X) stores the data source and alternate data source information.

Note: If there is an invalid mapping of data sources, the system displays an error message during search execution.

Modifying the Release/Data Source Map Table

This section 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 EnterpriseOne 8.10 version), the software converts the data for you when you are using a release of 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.

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. For 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.

Note: 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 EnterpriseOne tables only. Non-EnterpriseOne tables should be in a separate data source with decimal shift turned off. If you bring in a EnterpriseOne table as non-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-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 *IBM i* and Oracle data sources.

Note: If you select **I - DB2 for Linux, UNIX, and Windows database** on *IBM i* as the data source type, you can select either LOB Data Type Support or *IBM i* BLOB Support, but not both.

IBM i BLOB Support

The setting for the *IBM i* Binary Object (BLOB) Data source indicates whether a data source may contain EnterpriseOne BLOB Tables.

Understanding Data Source Reports

The 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 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	<p>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)</p> <p>In the Work With Batch Versions - Available Versions form, enter R98611 (Data Source Master Report) or R986112 (Data Source Master Comparison) in Batch Application.</p>	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.

Batch Versions - Work With Batch Versions - Available Versions

✓ 🔍 + 📄 🗑️ ✕ ⌵ Row 📄 Form ⚙️ Tools 🔄 One View

Batch Application: Object Configuration Copy Web and Client
 Read Only Report (Y/N):

Records 1 - 1

Version	Version Title	User	Last Modified	Security	Description	Client Platform
XJDE0001	Object Configuration Copy	CB5930119	03/15/2001	0	No Security	

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 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

OK Cancel Delete Up Down
✓ ✗ 🗑️ ^ v

	Operator	Left Operand	Comparison	Right Operand
<input type="checkbox"/>	Where	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Add Row](#)
[Advanced](#)

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 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	Used to run the Verify Object Configuration Mappings report.

Form Name	FormID	Navigation	Usage
		Source Management Reporting, Verify OCM. (P98305).	

Running the Verify Object Configuration Mappings Report

Access the Work with Batch Versions - Available Versions form.

Batch Versions - Work With Batch Versions - Available Versions

✓ 🔍 + 📄 🗑️ ✕ ⌵ Row ⌵ Form ⚙️ Tools 🔄 One View

Batch Application: Object Configuration Copy Web and Client

Read Only Report (Y/N):

Records 1 - 1

Version	Version Title	User	Last Modified	Security	Description	Client Platform
XJDE0001	Object Configuration Copy	CB5930119	03/15/2001 0	No Security		

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 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 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

OK Cancel Delete Up Down



	Operator	Left Operand	Comparison	Right Operand
<input type="checkbox"/>	Where	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Add Row](#)

[Advanced](#)

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.

5 Understanding and Setting Up Environments

Understanding Environments

This section provides an overview of environments and discusses:

- Environment definitions
- Environment table relationships

See Chapter 4 "Using Environment Director to Copy an Environment to a New Environment" for a more automated approach to setting up environments.

Understanding Environments

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 Object Configuration Manager (OCM) mappings.

EnterpriseOne has these types of environments:

- Distributed data and distributed logic environments.
Distributed data and logic environments determine where data resides and where application processing occurs.
- Group of object environments.
Group of object environments control which set of objects is used, such as production, development, or pristine.

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 EnterpriseOne. The environments you assign to each user in the Library List Control table are validated at startup against the 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.

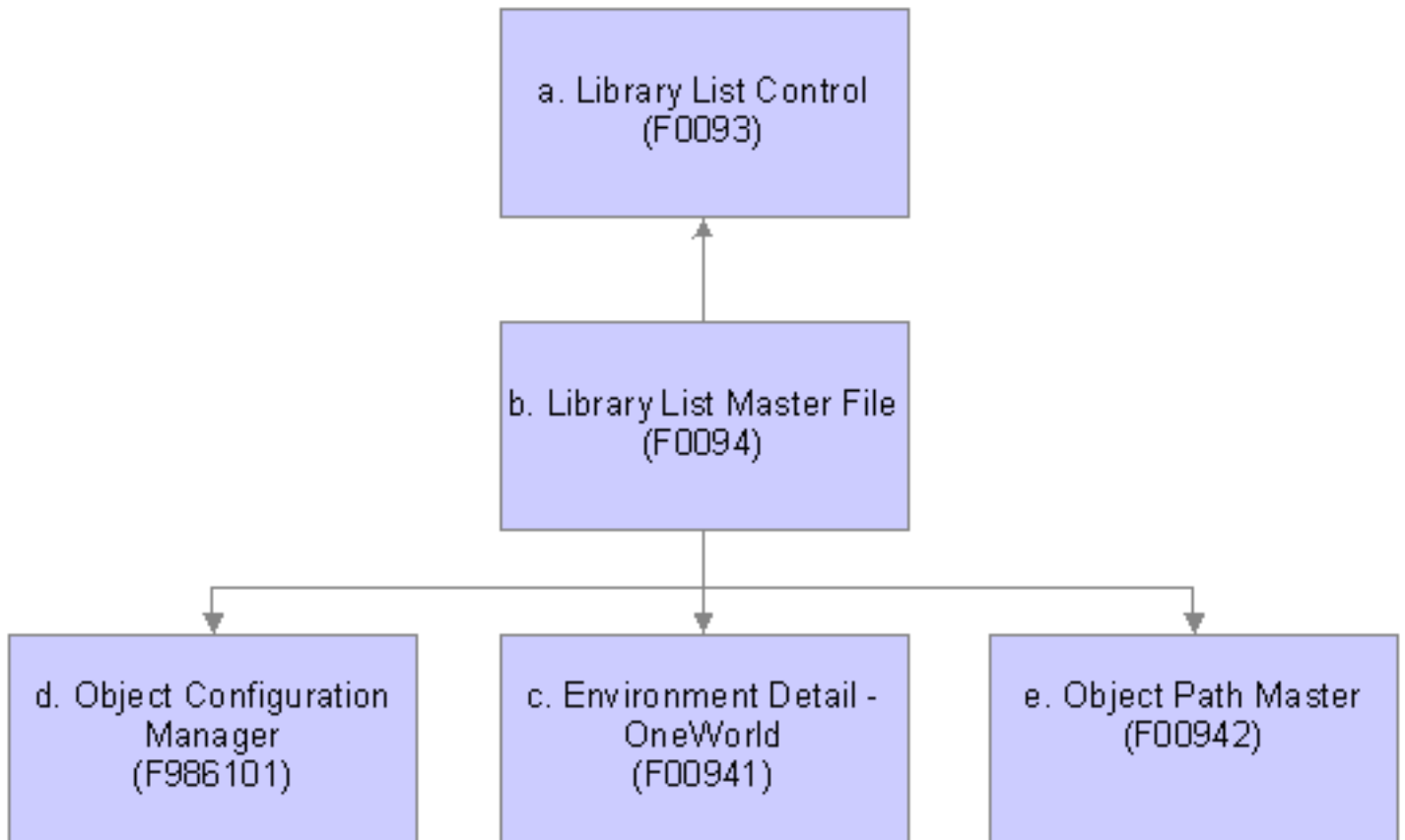
Object Path Master File Table (F00942)

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:



Setting Up Environments

This section provides an 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 if 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 Director

✕ ⚙ Next ⏪ Form ⚙ Tools

Welcome to the Environment Director!

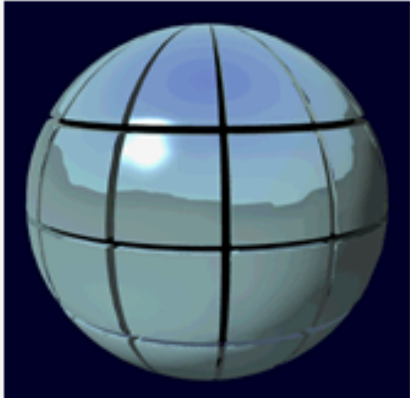
The Environment Director will step you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.

Click Cancel at any time to exit the Environment Director.

Click Next to begin.

New Environment Copy Environment

Existing Environment



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 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 EnterpriseOne objects which will be used to keep track of sets of objects and their locations within 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

✓ ✗ ⚙ Tools

Enter the data source to be used to create Object Librarian Object Configuration Mappings for the new Environment.

OL Data Source

OL Data Source

Identify the data source.

Setting Up Color Coded Environments (Release 9.2.4)

You can set up environments to be color coded for when you need a clear visual distinction as to which environment you are logged into. This enables you to quickly identify non-production environments by color.

The UDC 98/CO allows you to list environments and identify the color scheme for that environment. If no environment is listed the default color scheme is retained.

Note: Color coded environments is not compatible with Internet Explorer 11.

Setting Up Color Coded Environments

To Set up color coded environments:

1. Log into JD Edwards EnterpriseOne.
2. Enter UDC in the Fast Path.
3. Enter 98 in the Product Code field.
4. Enter CO in the User Defined Codes field.
5. Click find to verify the table exist.
6. Click the Add button.
7. Enter the environment you want a different color scheme for in the Codes column.
8. Enter the hex code for the color in the description 1* field.
9. Enter the next environment and color.
10. Click the OK button
11. Close User Defined Codes application.
12. Log out of JD Edwards EnterpriseOne.
13. Log into an environment with an associated color scheme.

Note: If the colors do not appear, clear the server cache with Cache Management in Server Manager.

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	<p>In Solution Explorer, navigate to System Administration Tools, System Administration Tools, Environment Management, Environment Master(P0094).</p> <p>In the Work With Environments form, click Find.</p> <p>Highlight the row you want and click Select.</p> <p>In the Row menu, click Copy Environment.</p>	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 EnterpriseOne tables.

Access the Work With Environments form.

Copy Environment

✓ ✗ ⚙ Tools

Existing Environment

Enter name of new environment :

New Environment

Copy *Public Records Only

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 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 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 Code from 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

✓ ✗ ⚙ Tools

Path Code	<input type="text" value="DV910"/>
Description	<input type="text" value="E910 Development Path"/>
Location	<input type="text" value="DNVMIEDP13"/>
Server Share Path	<input type="text" value="E910"/>
Status Code	<input type="text" value="1"/> <input type="button" value="▼"/> <i>In Production</i>
Merge Option	<input type="text" value="1"/> <input type="button" value="▼"/> <i>Merge</i>
Release	<input type="text" value="E910"/> <input type="button" value="▼"/> <i>Release 9.1</i>
Deployment Data Source	<input type="text" value="Central Objects - DV910"/>
UNC Flag	<input type="text" value="Y"/>
Cumulative Description	<input type="text"/>

XML Specifications Y/N

Local DB

OEE SSE Other

Path Code

Specify a pointer to a set of EnterpriseOne objects which will be used to keep track of sets of objects and their locations within 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 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.
 - o For logical data sources, you will probably want Business Data Test, Central Objects Test, EnterpriseOne Local Test, and Control Tables Test.
 - o 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	<p>In Solution Explorer, navigate to System Administration Tools, System Installation Tools, Custom Installation Plan (P98240).</p> <p>In the Work With Installation Plans form, select your plan and expand it.</p> <p>Select the environment and click Select.</p> <p>In the Row menu, click Copy Environment.</p> <p>Save the environment you added.</p>	Used to update the server map tables with the new environment.
Work with Locations and Machines	W9654AA	<p>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 expand EnterpriseOne Servers. Select the server you want to reuse. Select Generate Server Map from form menu.</p> <p>Select Generate Svr. Map from the Form menu</p>	Used to revise server map tables.

Adding a New Path Code on the Enterprise Server

To add a new path code on the EnterpriseOne Server:

1. Sign on to the 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\E910\ddp\PS910 (copy PS910 to TS910)

Unix Path Example

This is an example of a Unix path:

/u01/jdedwards/E910/PS910 (copy PS910 to TS910)

IBM i Path Example

You can also map a network drive to the root directory and do the first step from the deployment server.

1. MKDIR TS910 CPY OBJ('/PS910/*') SUBTREE(*ALL) OWNER(*KEEP)
2. Then copy the path code library: ADDLIB E910SYS CPYLIB FROMLIB(PS910) TOLIB(TS910) 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	<p>System Administration Tools, System Administration Tools, Environment Management, Environment Master (P0094).</p> <p>In the Work With Environments form, click Find.</p> <p>Highlight the row you want and click Select.</p> <p>Under File, click Delete.</p>	Used to delete an environment.

Form Name	FormID	Navigation	Usage
		In the Confirm Delete message, click OK.	

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

✓ 🔍 + 🗑️ ✕ ⋮ Row 🔄 Form ⚙️ Tools 🔄 One View

Display Only JD Edwards Environments

Records 1 - 10 > ⌂

<input type="checkbox"/>	Environment Name	Description	Path Code	Release	Just In Time Installation	Developer (Y/N)	WAN Configured	Detached Mode	Default Environment
<input type="checkbox"/>	DEP910	E910 Deployment Server Enviro	PLANNER	E910	N	Y			
<input type="checkbox"/>	DV910	E910 Development Environment	DV910	E910	Y	Y			
<input type="checkbox"/>	DV910C1	E910 SQE Constants C1	DV910	E910	Y	Y	1		
<input type="checkbox"/>	DV910C2	E910 SQE Constants C2	DV910	E910	Y	Y	1		
<input type="checkbox"/>	DV910C3	E910 SQE Constants C3	DV910	E910	Y	Y	1		
<input type="checkbox"/>	DV910C5	E910 SQE Constants C5	DV910	E910	Y	Y	1		
<input type="checkbox"/>	DV910C7	E910 SQE Constants C7	DV910	E910	Y	Y	1		
<input type="checkbox"/>	INDEV910	E910 Save Location	INDEV910	E910	N	Y			
<input type="checkbox"/>	JDV910	E910 Development Environment	DV910	E910	N	Y	1		
<input type="checkbox"/>	JPY910	E910 Prototype Environment	PY910	E910	Y	Y	1		

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 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 EnterpriseOne objects, and is used to keep track of sets of objects and their locations within 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.

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 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 (DV910, PS910, PD910, PY910)	HTML server/WTS Environment (JDV910, JPS910, JPD910, JPY910)
UBE	Enterprise Server	Enterprise Server

Type	Base Environment (DV910, PS910, PD910, PY910)	HTML server/WTS Environment (JDV910, JPS910, JPD910, JPY910)
BSFN	LOCAL	Enterprise Server

This table defines the object name and data class.

Object Name	Data Class
F0094	S
F9860	O
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 - 910
ENV1	F9860	*PUBLIC	Object Librarian - 910

Environment	Object Name	User/Role	Data Source
ENV1	F9200	*PUBLIC	Data Dictionary - 910
ENV1	F0004	*PUBLIC	Control Tables - ENV1
ENV1	F983051	*PUBLIC	Versions - PY910
ENV1	GT92002	*PUBLIC	Data Dictionary - 910
ENV1	GT9860A	*PUBLIC	Object Librarian - 910
ENV2	DEFAULT	*PUBLIC	Business Data - ENV2
ENV2	F0094	*PUBLIC	System - 910
ENV2	F9860	*PUBLIC	Object Librarian - 910
ENV2	F9200	*PUBLIC	Data Dictionary - 910
ENV2	F0004	*PUBLIC	Control Tables - ENV2
ENV2	F983051	*PUBLIC	Versions - DV910
ENV2	GT92002	*PUBLIC	Data Dictionary - 910
ENV2	GT9860A	*PUBLIC	Object Librarian - 910

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

Source	UBE Copy Program
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 i*

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/e910.

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 i Platform

To set up database components for *IBM i*:

1. Sign onto the enterprise server as a user with authority to create libraries.
2. Create these libraries:
 - o XXXDTA (substitute your business data library)
 - o XXXCTL (substitute your control tables library)
 - o XX910 (substitute your path code)
 - o XX910FA (substitute your target package name)
 - o COXX910 (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:
 - o @set NEWENV=YES
 - o @set NEWPATHCODE=YES
 - o @set UPATHCODE= xx910 (substitute your path code name)
 - o @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\E910\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:
 - o export NEWENV=YES
 - o export NEWPATHCODE=YES
 - o export LPATHCODE=xx910 (substitute your path code name)
 - o export LENV=xxx (substitute your environment prefix)
3. Verify the settings in the ce_CallInstall.sh file: export DATABASE_INSTALL_PATH=/u01/jdedwards/e910/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:
 - o @set NEWENV=YES
 - o @set NEWPATHCODE=YES
 - o @set UPATHCODE=XX910 (substitute your path code name - uppercase)
 - o @set LPATHCODE=xx910 (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\E910\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= xx910 (substitute your path code name - lowercase)
 - export UPATHCODE=XX910 (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/e910/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\E910\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= xx910 (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 P980WSEC, select the form exit Add System User.
2. On Work with System Users, enter the appropriate data source owner (for example, DV910, PY910 or PD910) 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 an 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

Oracle recommends that you use the Environment Director and not the 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:

```
rmmdir /S /Q d:\JDEdwards\e910\XX910(substitute your target path code directory)
```

You can use the Unix command:

```
rm -Rf /u01/jdedwards/e910/XX910
```

On *IBM i*, 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	W9861150	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. 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.

Form Name	FormID	Navigation	Usage
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.
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.

Copy Environment / Packages

   Next  Form  Tools

This application can be used to copy Data, Central Objects, Path Code, Packages 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.

Copy definitions from System to Planner

Copy Environment definitions

Environment Copy Mode

Copy to defined environment

Environment defined in Planner database. Environment Master, Path Code Master and Data Source Template records exist in planner. May not have been installed.

Create / copy to New Environment

Environment not defined in planner databases

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

✕ ⚙ Next ⏪ Form ⚙ Tools

Welcome to the Environment Director!

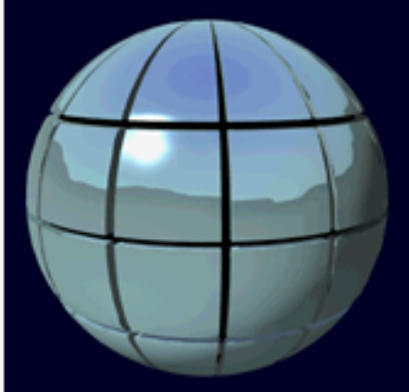
The Environment Director will step you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.

Click Cancel at any time to exit the Environment Director.

Click Next to begin.

New Environment Copy Environment

Existing Environment



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

  Back  Next  Form  Tools

Choose Director to step through the director. The director 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.

Choose Express to 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.

- Director
- Express

Express Information

New Environment

Environment Name

Description

Share Existing Path Code

Path Code Name

Enterprise Servers

[Click here to select the enterprise servers that will be configured for the new environment.](#)

Director

The Director mode enables you to copy your Central Objects and Business Data from one path code to another, and your Control Tables and Path Code from one environment to another. It also sets up the environment and path code definitions (EnterpriseOne architecture data). 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 sets up the EnterpriseOne architecture definition (OCM, Path Code Master and so on) for the new environment, but does not copy any data. 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 of that environment are presented as defaults on this form.

Access the Environment Properties form.


Environment Properties

  Back  Next  Form  Tools

Specify properties for the new environment. If an existing environment is being copied, the properties for that environment are presented as defaults.

The Prefix is used for creating data sources. For example, prefix "PROD" will generate "Business Data - PROD" and "Control Tables - PROD".

JAS is used for a Java environment. This affects OCM mappings for logic.

Environment Name	<input type="text"/>
Description	<input type="text"/>
Prefix for data	<input type="text"/>
Release 	<input type="text"/>
<input type="checkbox"/> JAS/WTS Environment	

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.

HTML Server/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

  Back  Next  Form  Tools

Select **Share an Existing Path Code** to share an existing path code with another environment.

Select **Copy an Existing Path Code** to copy an existing path code for use by the new environment.

Share an Existing Path Code

Copy an Existing Path Code

Existing Package Name

New Package Name

[Check Disk Space](#)

Share an Existing Path Code

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

Copy an Existing Path Code

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

Existing Package Name

The existing package name is the source package in your source path code.

New Package Name

The new package name is the target package in the target path code that you are creating.

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

  Back  Next  Form  Tools

Specify the properties for the new path code. The properties for the existing path code are used as defaults for the new path code.

Path Code *

Description

Release

Release *

Cumulative Description

Check In

Location *

Server Share Path *

UNC Flag

Deployment Data Source *

Options

Status Code

Merge Option

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

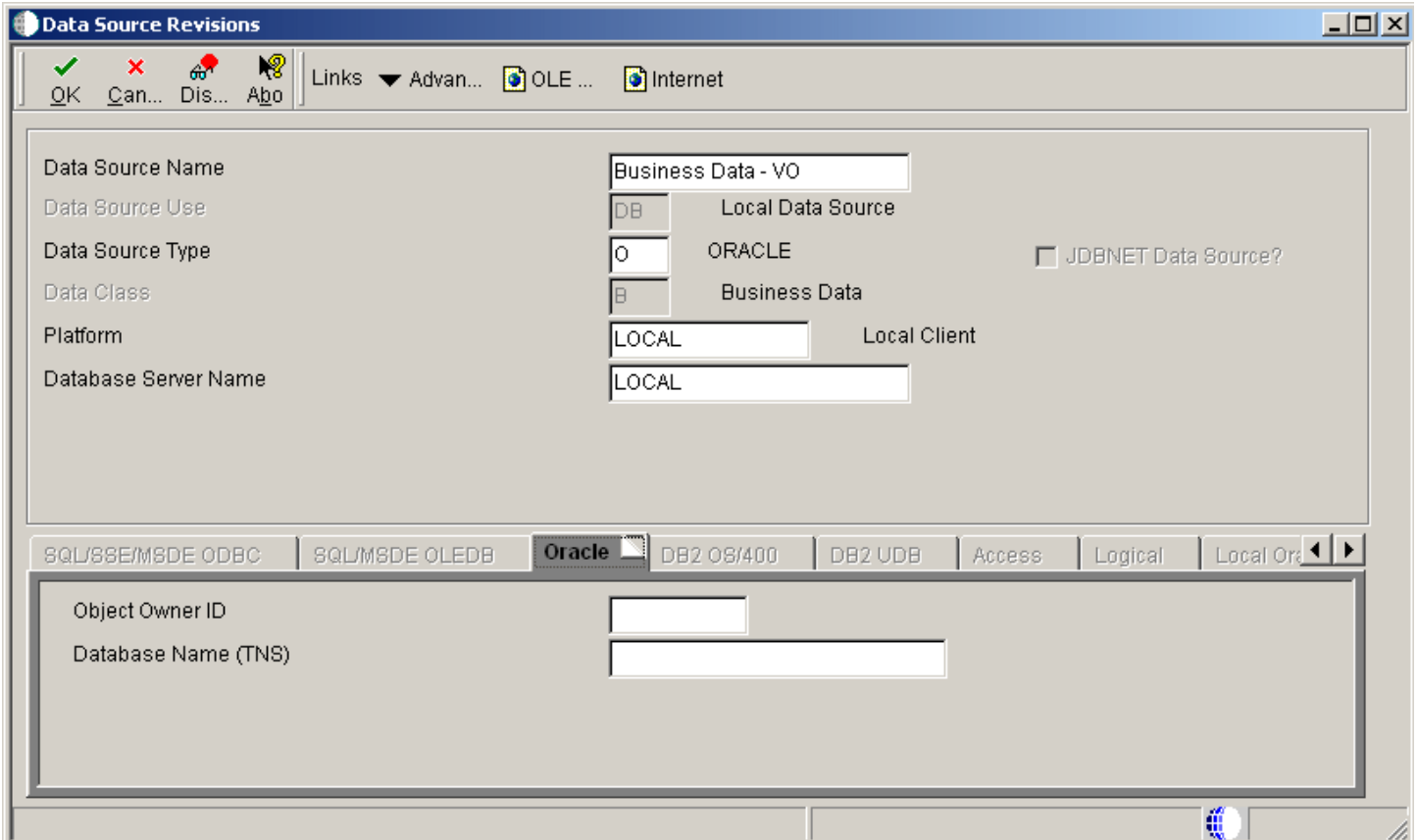
Enter a code to determine the status of the software in the development cycle.

Merge Option

Select this option to determine whether a customer's object will be merged in with the 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 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 from clients.

Machines

The Machines form is used to select the enterprise servers where the copied environment will run. 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 from one server to another, you must do the copy yourself using operating system utilities.

For example:

- *IBM i* - 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 Tools

Select the Enterprise Servers where the new environment will run. EnterpriseOne will use this information to populate the server map data sources on these servers.

[Click here to automatically select the enterprise servers from the existing environment.](#)

Filter Available Enterprise Servers

Machine Key * Find
Port Number

The screenshot displays two side-by-side panels for managing enterprise servers. The left panel, titled 'Available Enterprise Servers', contains a list of two server entries: 'den60201jems - Port 6133' and 'den60201jems - Port 6033'. Each entry has a radio button to its left. The right panel, titled 'Selected Enterprise Servers', is currently empty. Between the two panels are two blue arrows: a right-pointing arrow at the top and a left-pointing arrow at the bottom, indicating the ability to move servers from the available list to the selected list and vice versa.

Machine Key

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

Port Number

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

Note: You must select at least one enterprise servers before clicking Next. To do so, click the enterprise server located in the left field below Port Number, and then click the Right arrow.

Data Script Generator

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

Access the Data Script Generator form.

Database Script Generator

✕ ⚙ Back ⚙ Next ⚙ Form ⚙ Tools

Operating System

Windows Unix iSeries

Database Type

Oracle UDB SQL Server

SQL Server Properties

Database Server Name

Database Server Path

Operating System

Select the operating system.

Database Type

Select the database type.

SQL Server Properties

Select the SQL server properties.

The Database Server Name and Database Server Path fields were used when selecting Windows and SQL. These fields can now be left blank and will have no affect on the environment copy process.

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 ⚙ Tools

Environment Data

<input checked="" type="checkbox"/> Load Business Data Click here to customize the Business Data data load.	<input checked="" type="checkbox"/> Load Control Tables Click here to customize the Control Tables data load.
--	--

Path Code Data

<input checked="" type="checkbox"/> Central Objects <input checked="" type="checkbox"/> Check Out Records	<input checked="" type="checkbox"/> Versions <input type="checkbox"/> ESU History
--	--

Directories

<input checked="" type="checkbox"/> Copy Path, Pkg Dir on Deployment Server <input checked="" type="checkbox"/> Copy Pkg Hdr / Dtl Recs <input type="checkbox"/> Copy Path Code, Pkg Dir on Enterprise Server

Metadata repository

<input type="radio"/> Copy Metadata Repository <input type="radio"/> Rename Metadata Repository <input checked="" type="radio"/> Don't Copy Repository	<input type="checkbox"/> Configure Pkg Manifest
--	---

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. All UBEs must run locally except for the server path code copy.

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.

Copy Path Code, Pkg Dir on Deployment Server (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. This also copies the package definition records and package ini files.

Copy Path Code, Pkg Dir on Enterprise Server (R9894003 XJDE0001)

This copies the path code and the package directories on the enterprise server. This UBE is submitted to the enterprise server.

Copy Metadata Repository (R9894005 XJDE0001)

This copies the Metadata Repository (the package specifications) from the source package to the target package. It will copy the repositories both locally and on the enterprise database.

Rename Metadata Repository (R9894005 XJDE0001)

This renames the tables in the target metadata repository. Click this option if you have copied the metadata repository on the server using the database commands.

Configure Package Manifest (R9894005 XJDE0002)

This configures the BLOB in the manifest table (F98770) in the target package repository. It will perform the configurations both locally and on the database server.

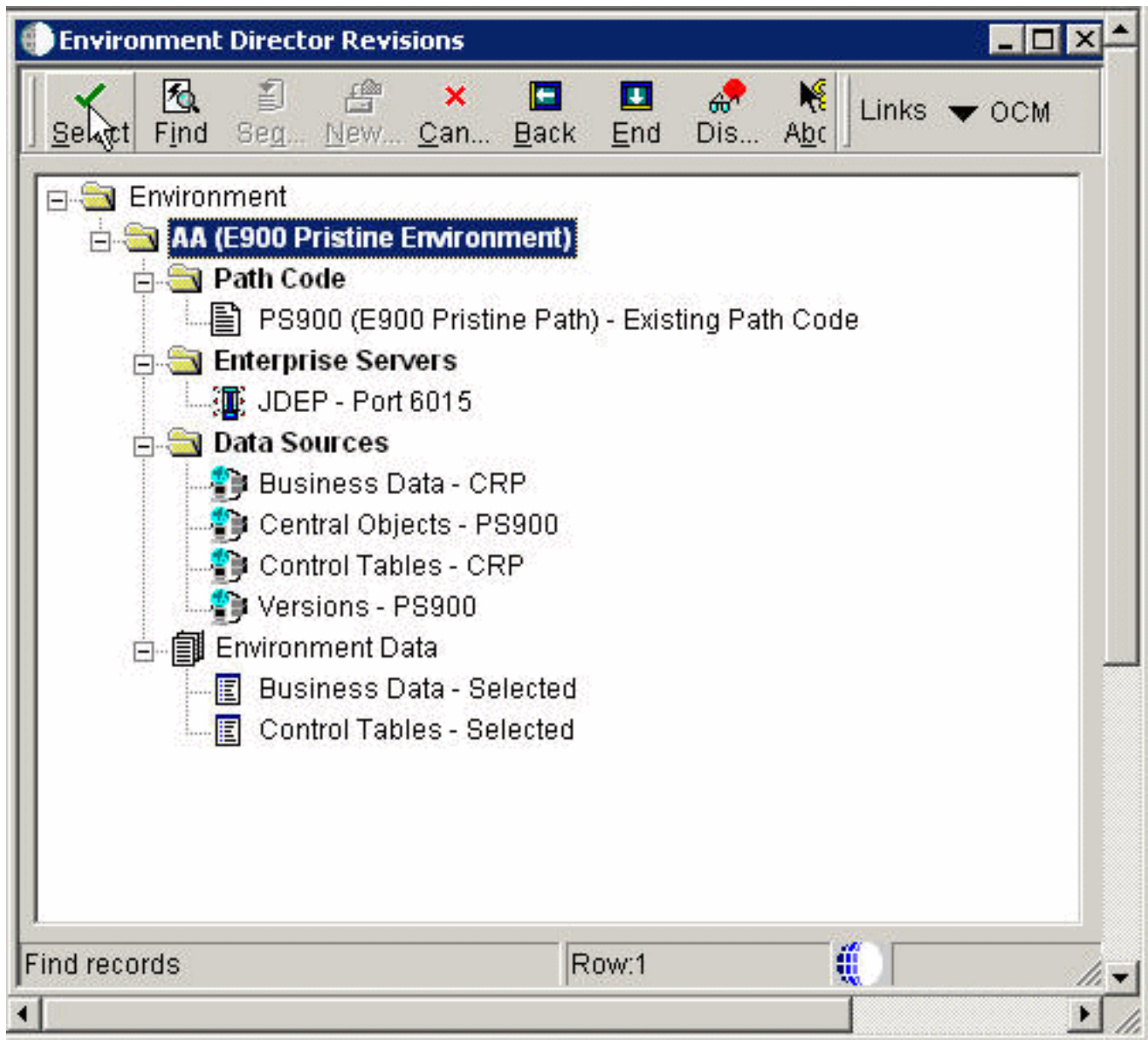
Environment Director Revisions

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.



Path Code

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

P984052 - [Table Conversion/Merge Log]

File Edit Preferences Row Window Help

Find Del... Close Seq... Ne... Dis... Abo... Links View ... OLE ... Internet

From: Start Date A: Start Time

To: Start Date B: Start Time

Plan Name	Target Environ	Prev Environ	Cnv Type	Object Name	Cnv Sts	Conversion Status	Start Date	Beg Time	Compl Date	End Time	Program Name
E910INSTALL	DV910		90	Load Data	2	Conversion Finished Nor	2/5/20	10:36:2	2/5/20	10:40:1	R98403
E910INSTALL	PY910		90	Load Data	2	Conversion Finished Nor	2/5/20	10:40:1	2/5/20	10:44:2	R98403
JM15592	DV910		1	F9200	2	Conversion Finished Nor	11/17/	14:43:2	11/17/	14:44:3	R989200P
JM15592	DV910		1	F0004	2	Conversion Finished Nor	11/17/	14:44:3	11/17/	14:44:4	R9600042
JM15592	DV910		3	F00165	3	Conversion Finished Abn	11/17/	14:44:4	11/17/	14:44:4	R98403XB
JM15592	DV910		3	F00165	3	Conversion Finished Abn	11/17/	15:31:5	11/17/	15:31:5	R98403XB
JM15592 - 1	DV910		3	F00165	3	Conversion Finished Abn	11/17/	15:36:1	11/17/	15:36:1	R98403XB
JM15592 - 2	DV910		3	F00165	3	Conversion Finished Abn	11/17/	15:38:2	11/17/	15:38:2	R98403XB
JM15592 - 3	DV910		3	F00165	3	Conversion Finished Abn	11/17/	16:27:1	11/17/	16:27:1	R98403XB
JM15592 - 4	DV910		3	F00165	3	Conversion Finished Abn	11/25/	09:31:4	11/25/	09:31:4	R98403XB
JM15592 - 5	DV910		3	F00165	3	Conversion Finished Abn	11/25/	09:53:3	11/25/	09:53:4	R98403XB
JM15592 - 6	DV910		3	F00165	2	Conversion Finished Nor	11/25/	10:55:3	11/25/	10:55:4	R98403XB
JM15592 - 6	DV910		6	F98710	2	Conversion Finished Nor	11/25/	10:56:1	11/25/	10:57:3	R98700
JM16532	DV910		1	F9200	2	Conversion Finished Nor	11/25/	11:37:2	11/25/	11:37:2	R989200P
JM16532	DV910		1	F0004	2	Conversion Finished Nor	11/25/	11:37:2	11/25/	11:37:3	R9600042

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. 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

✕ ⚙ Next ⏪ Form ⚙ Tools

Welcome to the Environment Director!

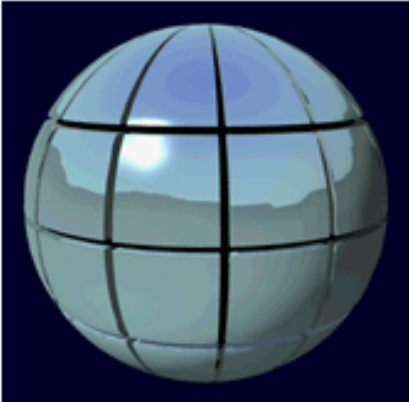
The Environment Director will step you through the process of defining the components of a new environment including environment, path code, enterprise servers, and data sources.

Click Cancel at any time to exit the Environment Director.

Click Next to begin.

New Environment Copy Environment

Existing Environment



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

  Back  Next  Form  Tools

Choose Director to step through the director. The director 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.

Choose Express to 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.

- Director
 Express

Express Information

New Environment

Environment Name

Description

Share Existing Path Code

Path Code Name

Enterprise Servers

[Click here to select the enterprise servers that will be configured for the new environment.](#)

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
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. EnterpriseOne will use this information to populate the server map data sources on these servers.

Access the Machines form.

Machines

✓ Form Tools

Select the Enterprise Servers where the new environment will run. EnterpriseOne will use this information to populate the server map data sources on these servers.

[Click here to automatically select the enterprise servers from the existing environment.](#)

Filter Available Enterprise Servers

Machine Key * Find

Port Number

The screenshot displays a configuration window with two main panes. The left pane, titled 'Available Enterprise Servers', contains a list of two server entries: 'den60201jems - Port 6133' and 'den60201jems - Port 6033'. Each entry has a radio button to its left. The right pane, titled 'Selected Enterprise Servers', is currently empty. Between the two panes are two blue arrows: a right-pointing arrow at the top and a left-pointing arrow at the bottom, indicating the ability to move servers from the available list to the selected list and vice versa.

Machine Key

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

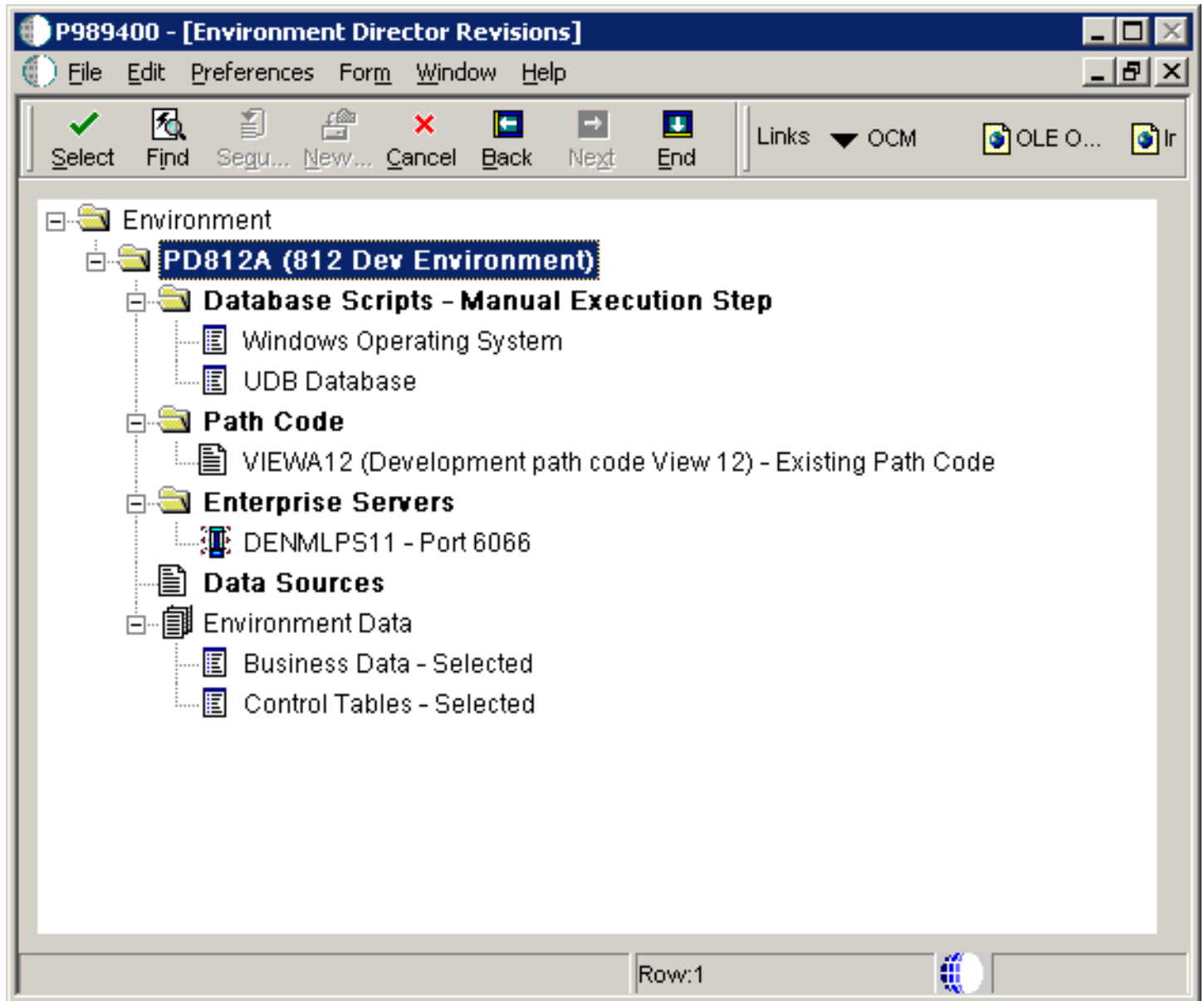
Port Number

Identify the port for a given instance of 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.



Path Code

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

Enterprise Servers

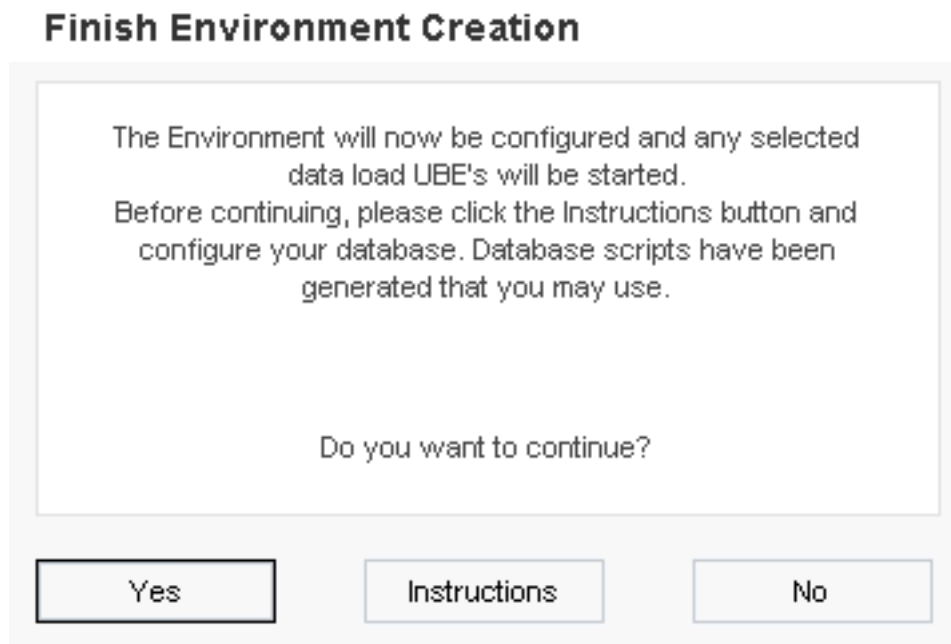
The names of the computers that receive data from the clients.

Data Sources

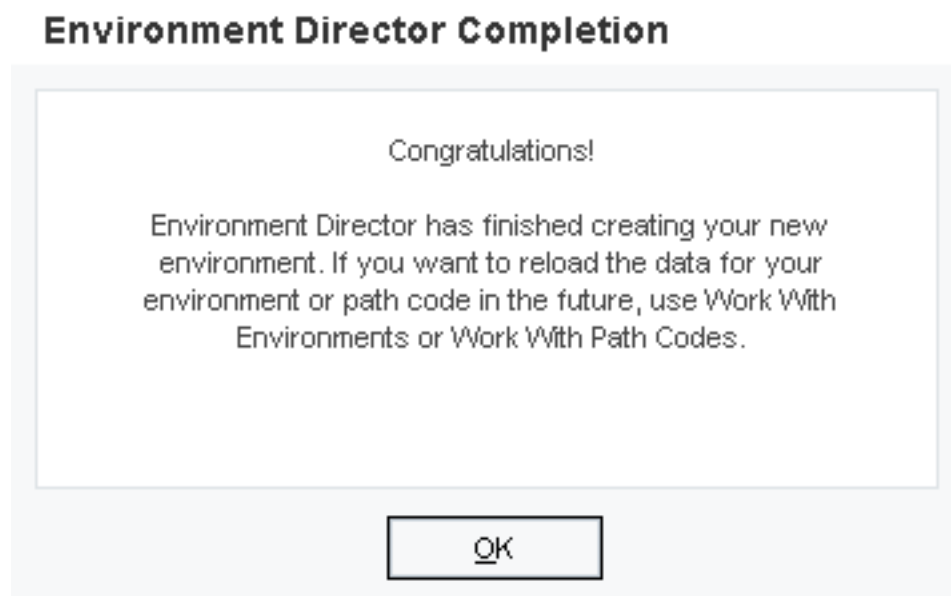
The names that identifies the data source.

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.



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



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 JDEPLAN 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

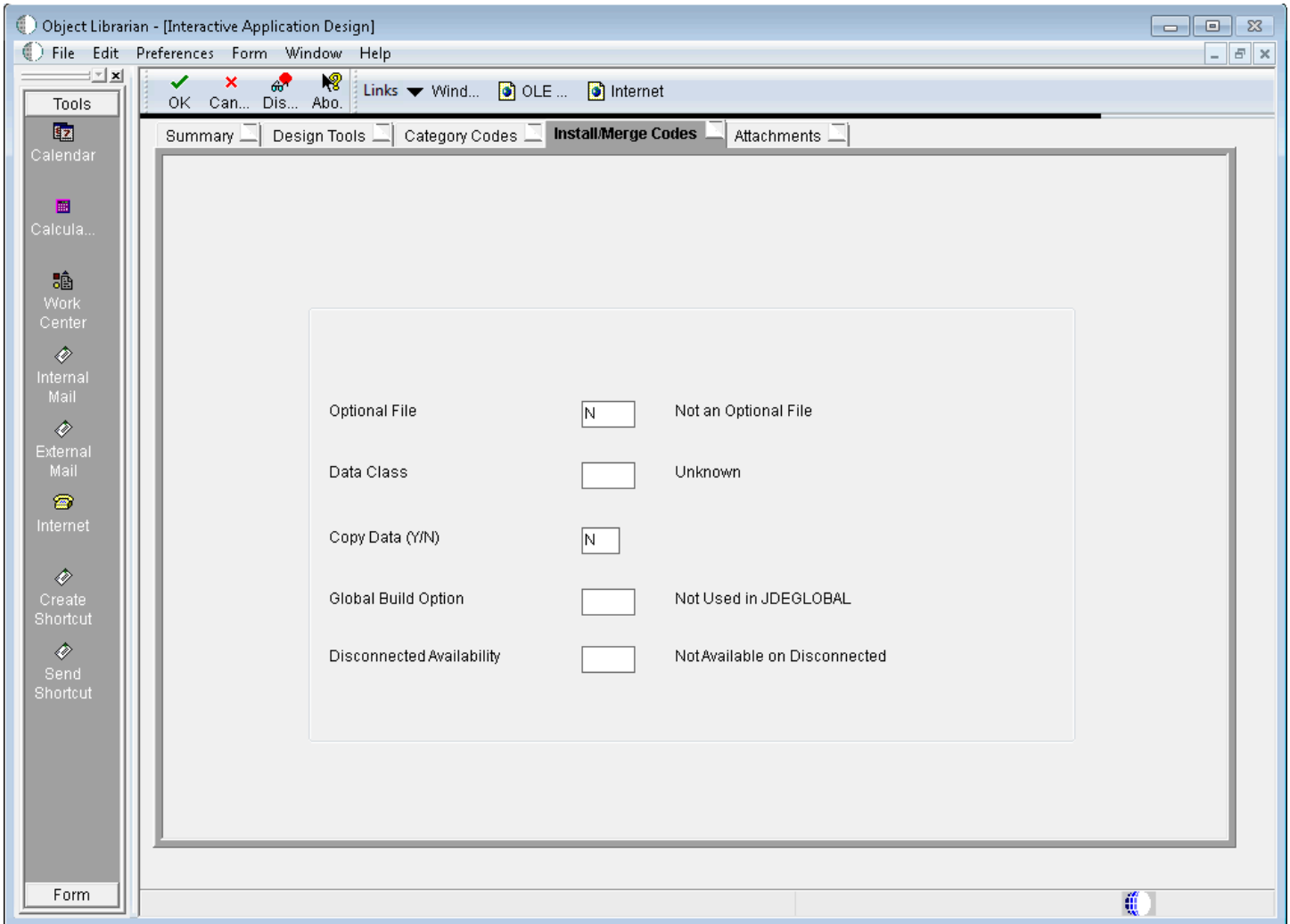
This section discusses how to use Object Management Workbench (P98220) to modify table data classes.

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.



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 an 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 an 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.

EnterpriseOne Text

Future Use.

Using On Track Planning Setup to Modify Table Data Classes

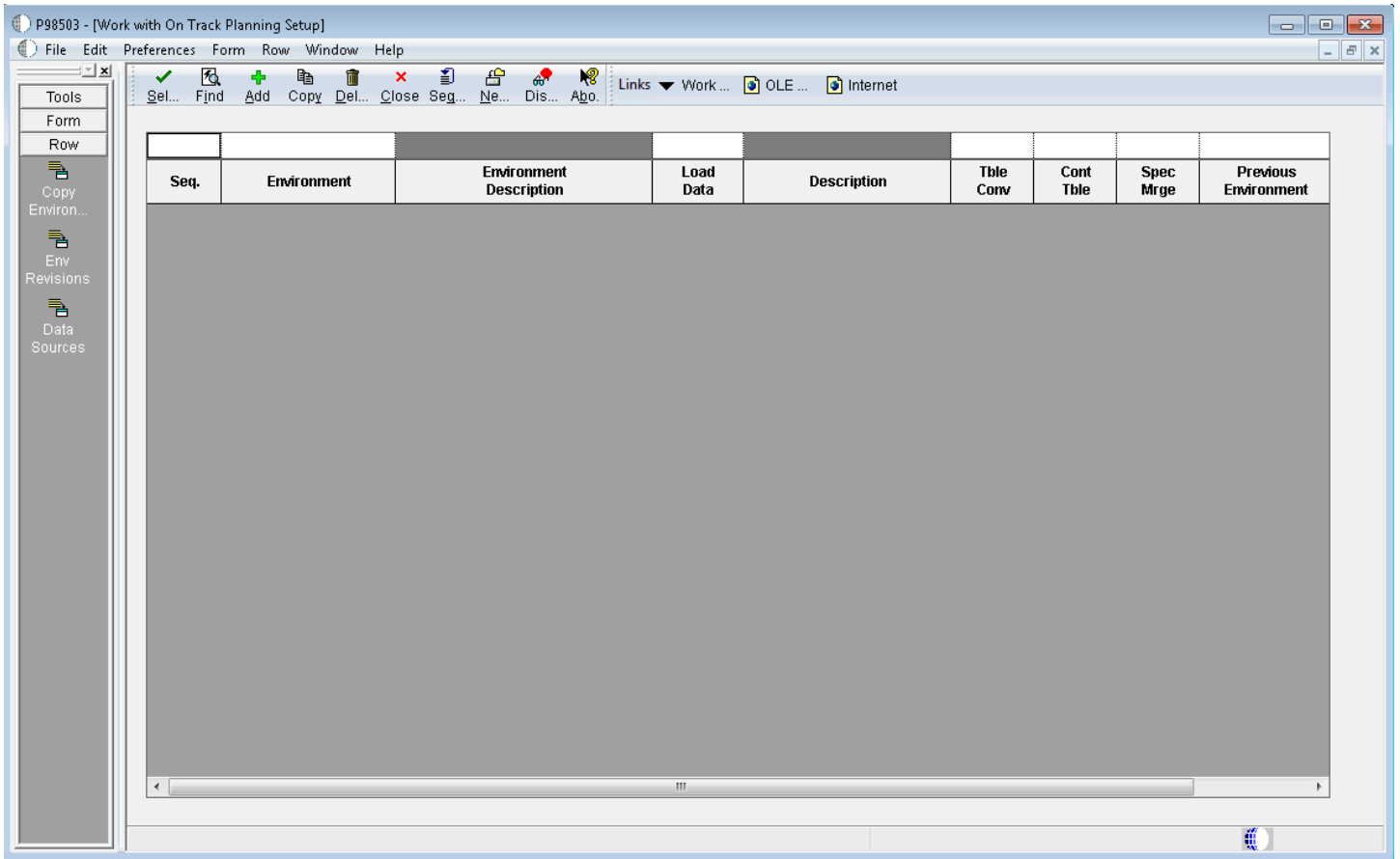
This section discusses how to use On Track Planning Setup (P985033) to modify table data classes.

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.



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 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.

6 Understanding and Working with Object Configuration Manager

Object Configuration Manager Overview

This section provides an 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.

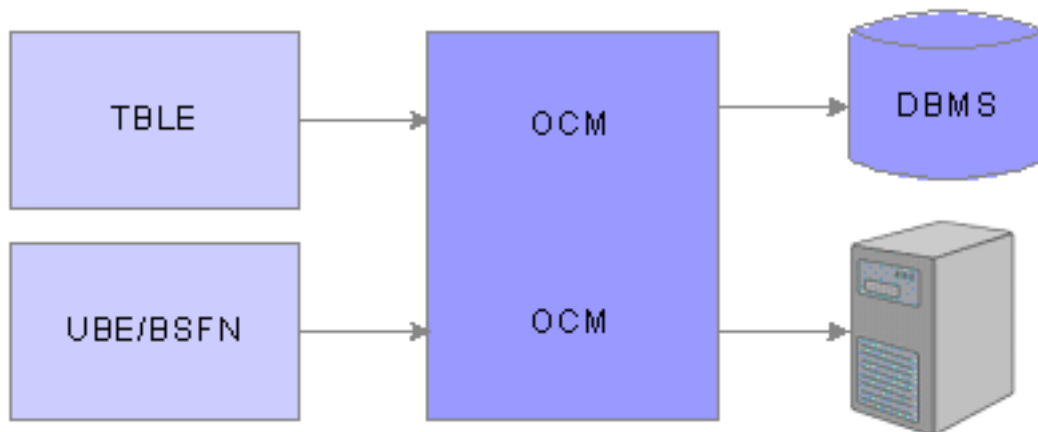
OCM has these characteristics:

- Data and logic resources can be located on any machine, and EnterpriseOne must know where each resource is located.
- OCM is the method of managing distributed data and distributed logic processing in an EnterpriseOne architecture 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.



Object Mappings

In 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.

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.

Note: 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 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 EnterpriseOne workstation while

simultaneously decreasing network traffic. This redeployment involves remapping objects using the 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).

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 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.

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 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, 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 PD910 or DV910.
- 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 PD910 (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?
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.

Search Sequence	Object Name	User or Role	Search Criteria

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, an 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.

Note:

- [JD Edwards EnterpriseOne Tools Runtime Administration Guide](#) .

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 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.

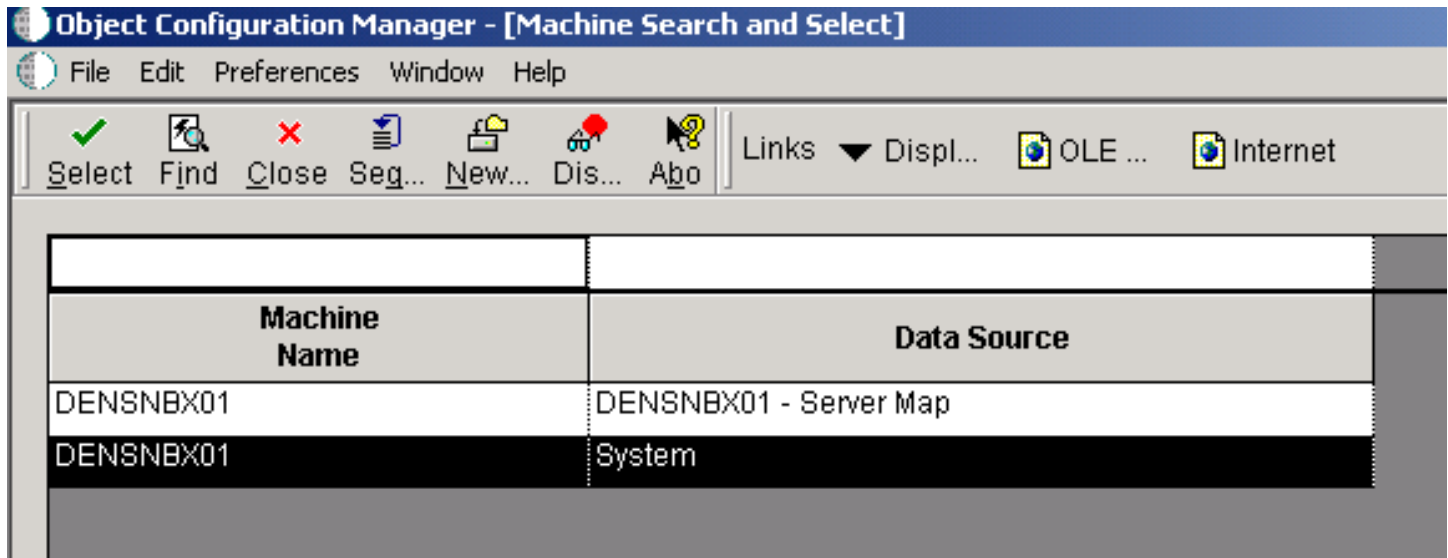
Setting Up Object Mappings for the Object Librarian Table

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 Name

A value that defines the logic host to EnterpriseOne.

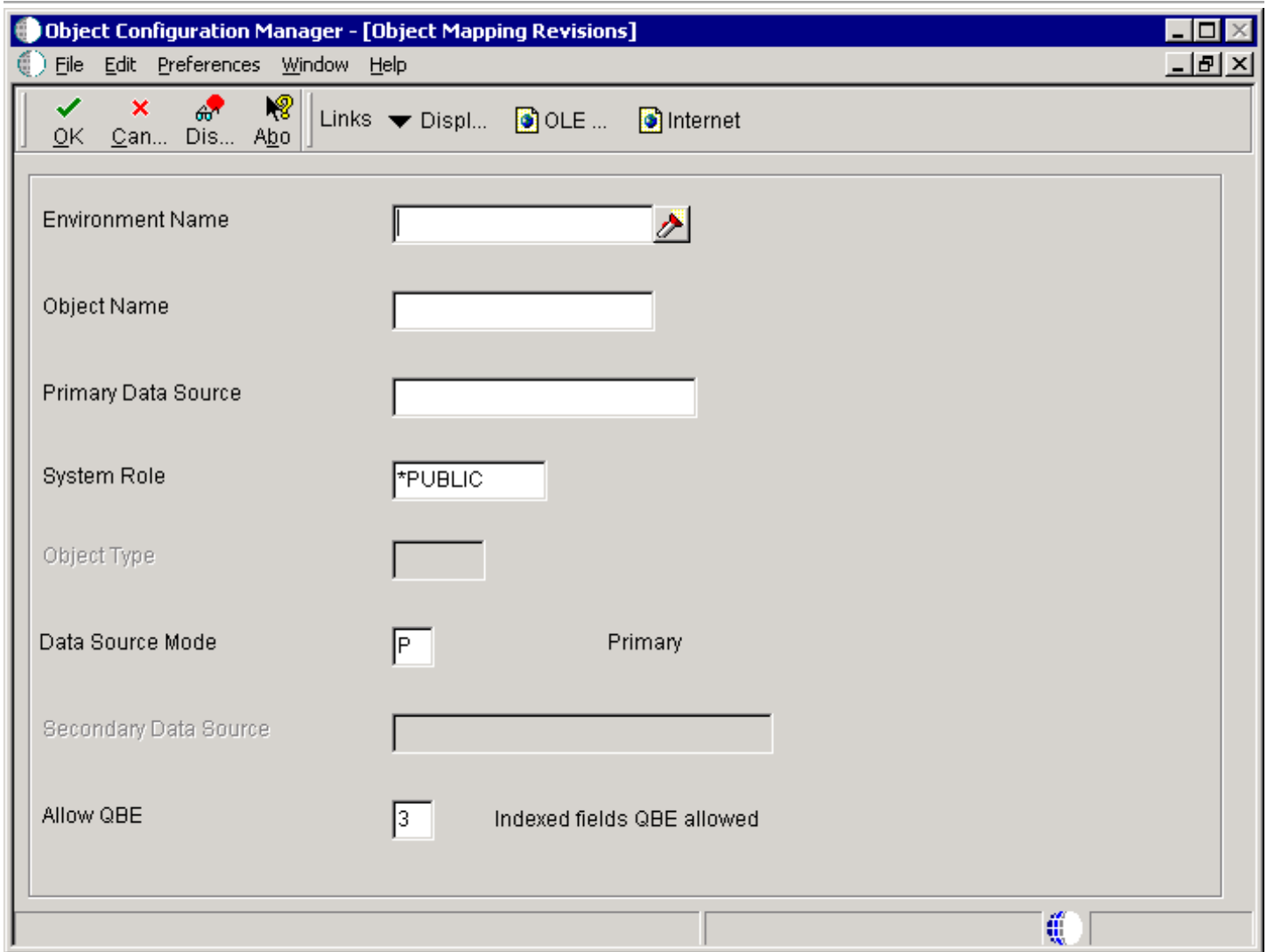
Data Source

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.



Environment

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

For 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. 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 an 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

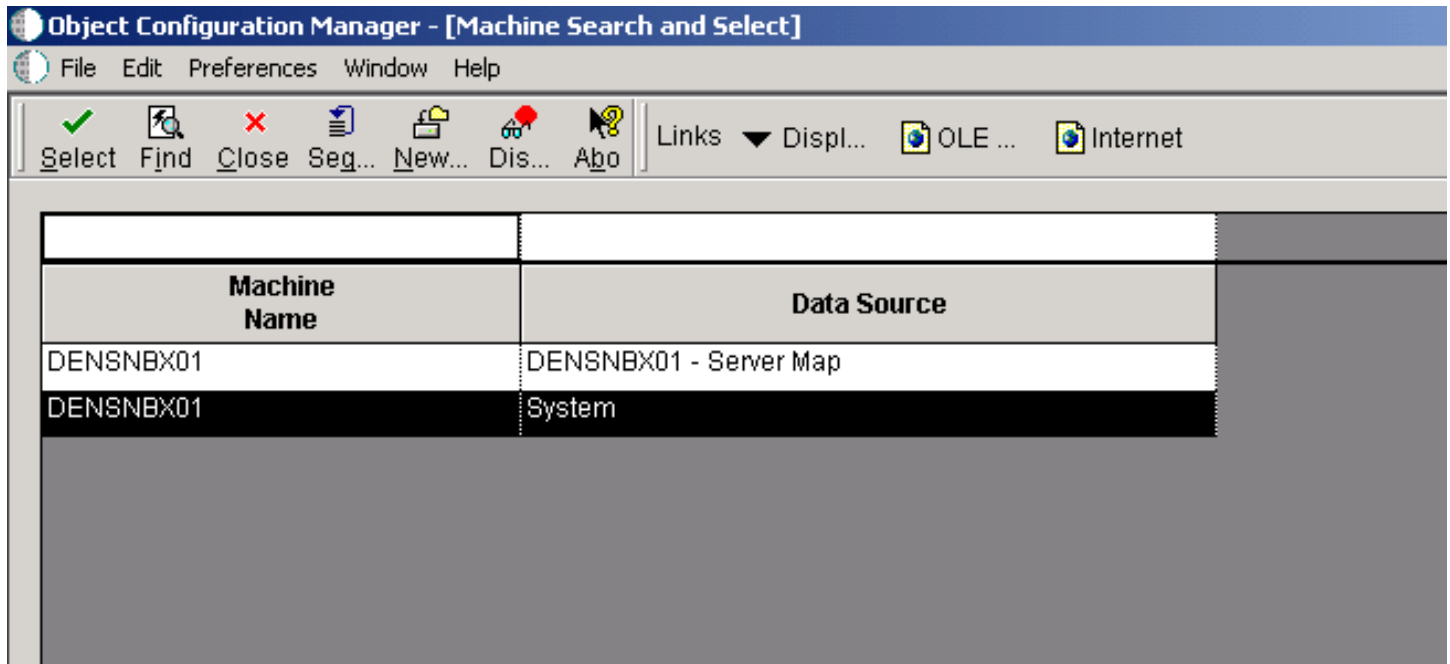
The Work With Object Mappings form is used to display data sources that have the OCM Data Source field checked on the Data Source Revisions form.

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 Name

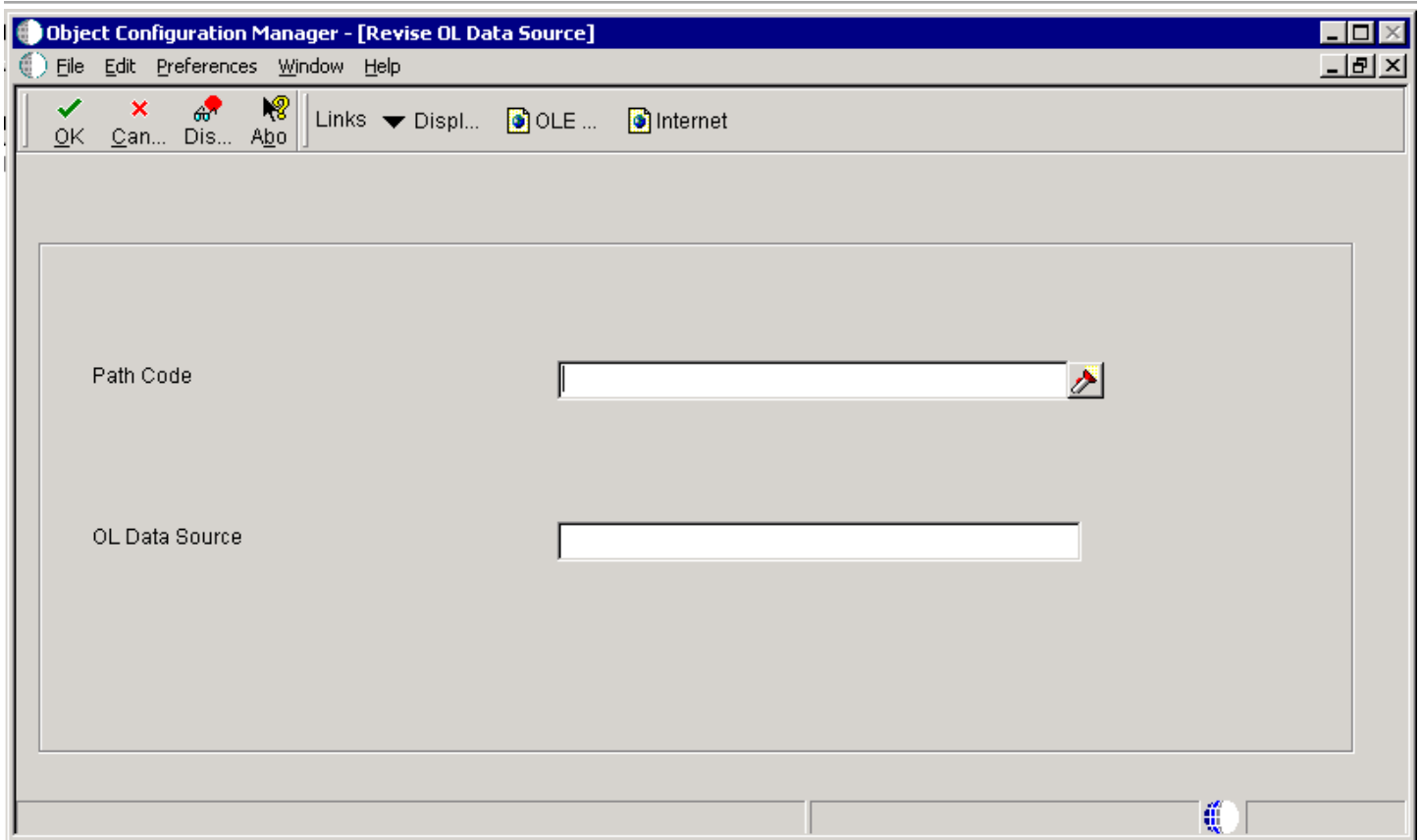
A value that defines the logic host to EnterpriseOne.

Data Source

The name that identifies the data source.

Revise OL Data Source

Access the Revise OL Data Source form.



Path Code

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

The screenshot shows a software window titled "Batch Versions - [Work With Batch Versions - Available Versions]". The window has a menu bar (File, Edit, Preferences, Form, Row, Window, Help) and a toolbar with icons for Select, Find, Add, Copy, Del..., Close, Seg..., New..., Dis..., and Abo. Below the toolbar, there are two input fields: "Batch Application" with the value "R986101A" and a "Create System Table Mappings" button, and "Read Only Report (Y/N)" with the value "N". Below these fields is a table with the following data:

Version	Version Title	User	Last Modified	Security	Description	Client Platform
XJDE0001	Map System Table for all environments	DEMO	10/13/1999	0	No Security	

At the bottom of the window, there is a status bar showing "Row:1" and a globe icon.

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.

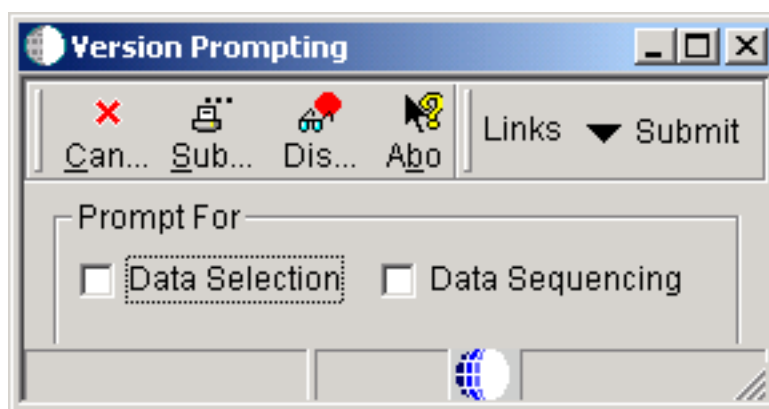
3 - Full Security Only the "Last Modified By" user can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

Description

A user defined name or remark.

Client Platform

A category code associated with the Versions List table for EnterpriseOne (F983051).



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 an 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.

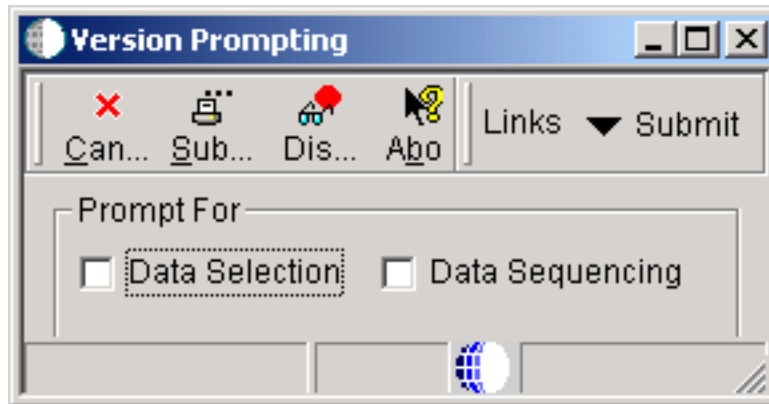
3 - Full Security Only the "Last Modified By" user can design, change processing option values, change detail values, check in, check out, install, copy, transfer, delete, or run the version. This is the default when adding a new version.

Description

A user defined name or remark.

Client Platform

A category code associated with the Versions List table for EnterpriseOne (F983051).



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 Parm from 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 Form menu.	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

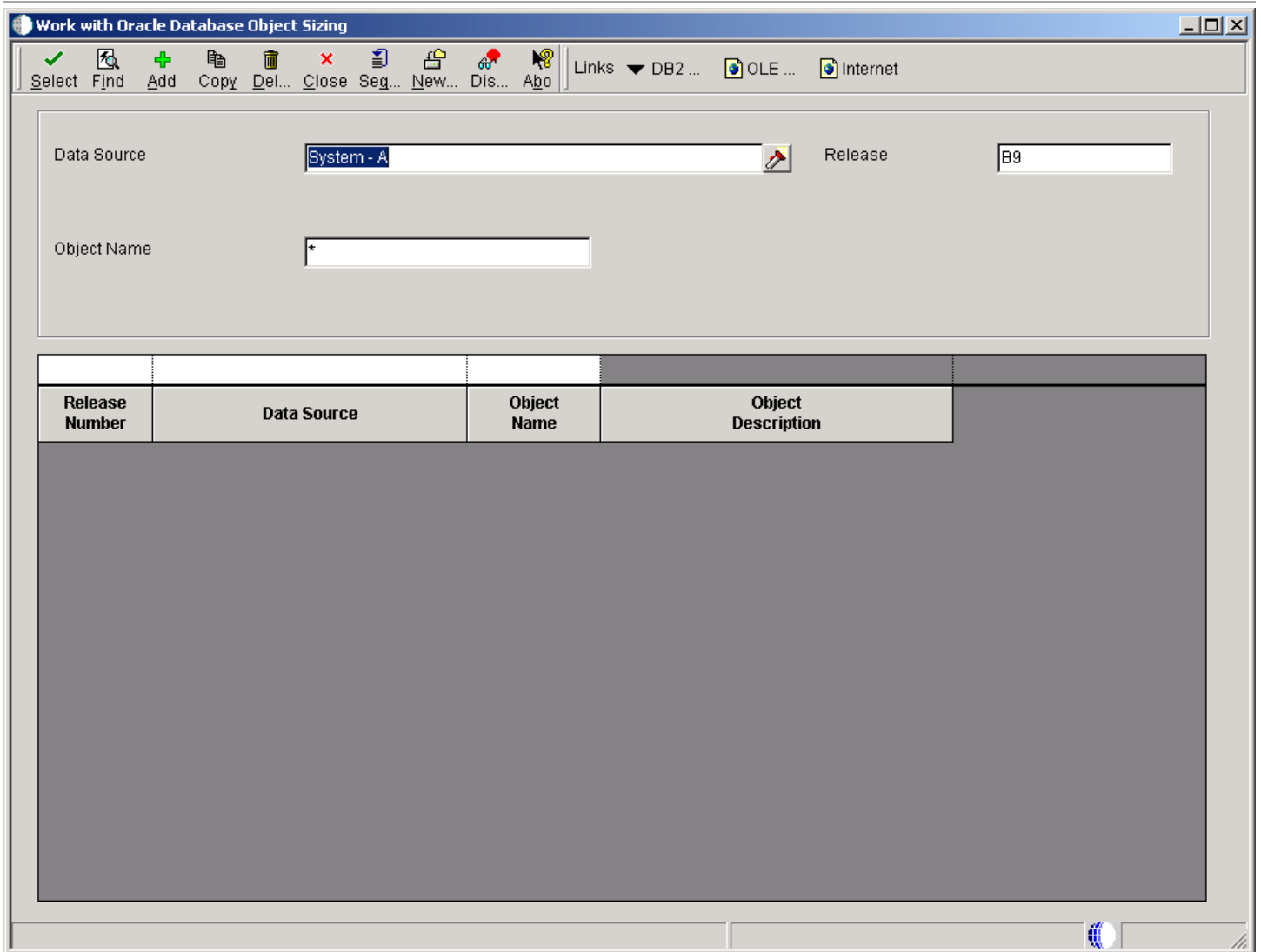
A value that defines the logic host to EnterpriseOne.

Data Source

The name that identifies the data source.

Work With Oracle Database Object Sizing

Access the Work With Oracle Database Object Sizing form.



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. 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.

The screenshot shows a window titled "P986110 - [Oracle Database Object Sizing]". The menu bar includes "File", "Edit", "Preferences", "Form", "Window", and "Help". The toolbar contains icons for "OK", "Cancel", "Dismiss", "Apply", "Links", "Override", "OLE ...", and "Internet".

The main area is titled "OLE Objects" and contains the following fields:

- Data Source:** System
- Release:** E812
- Object Name:** DEFAULT

Below these fields are two sections:

- Table Information:** Contains a field for **Table Space Name**.
- Index Information:** Contains a field for **Index Space Name**.

Data Source

Specify the name that identifies data source.

Object Name

Specify the name that identifies a system object. 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.

P986110 - [Revise Table and Data Source Overrides]

File Edit Preferences Window Help

OK Cancel Disconnect Abort Links Displ... OLE ... Internet

Release

Data Source

Object Name

Copy Data (Y/N)

Create Tables(1/0)

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 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 EnterpriseOne if your business follows the typical configuration: (This example includes only the PD910 (production) and TS910 (test) environments).

Environment	Object Name	Description	Data Source	Object Type
PD910	F00165	Generic Text Table	Business Data - Prod	TBLE
PD910	GT92002	Data Dictionary Glossary	Data Dictionary	GT
PD910	GT9860A	Object Librarian	Object Librarian	GT
PD910	GT9862A	Business Function Notes	Object Librarian	GT
PD910	GT98DSA	Data Structure Notes-Structure	Object Librarian	GT

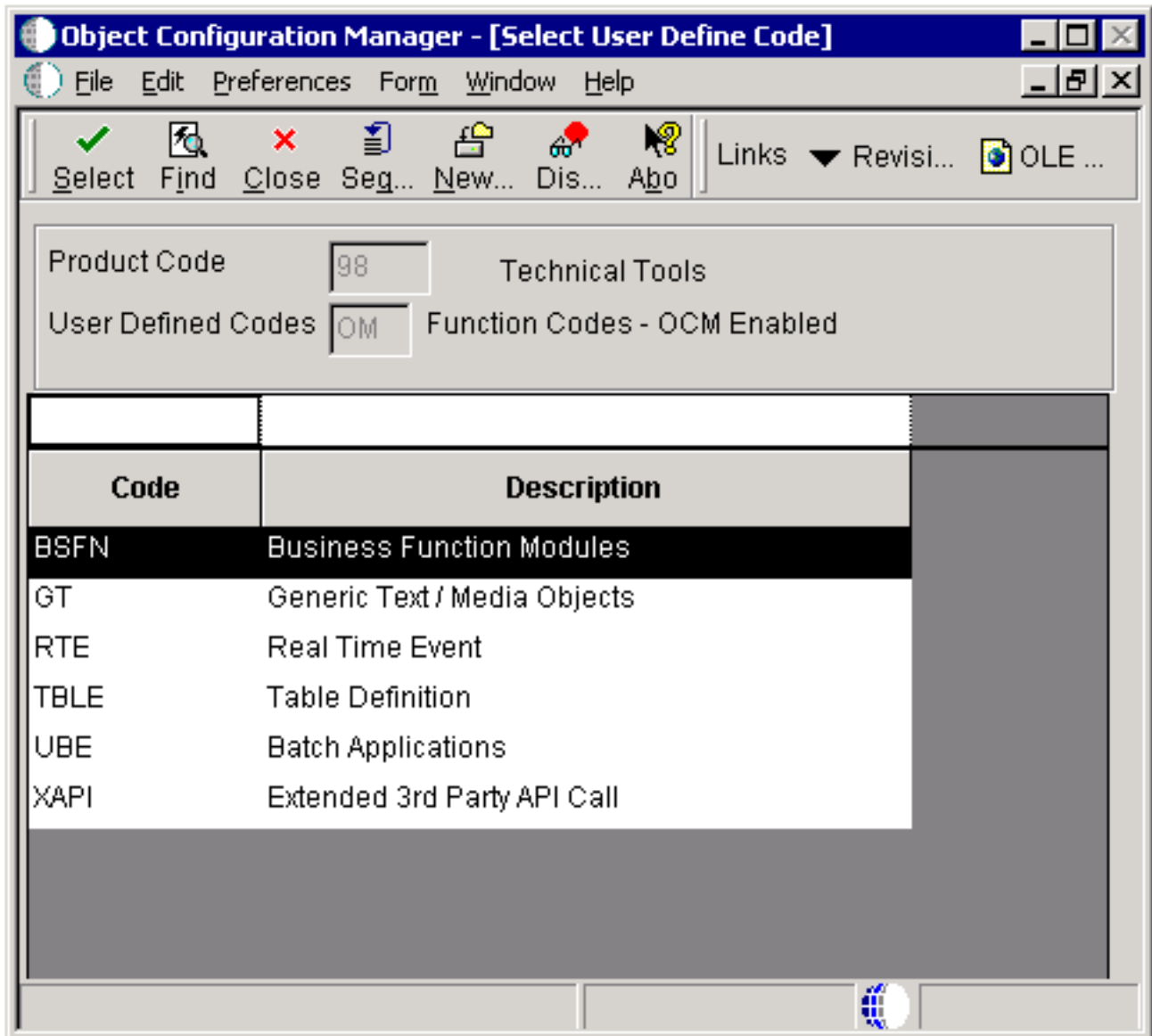
Environment	Object Name	Description	Data Source	Object Type
PD910	GT98DSB	Data Structure Notes-Structure and Item	Object Librarian	GT
PD910	GT98TMPL	Media Objects Templates	Object Librarian	GT
TS910	F00165	Generic Text Table	Business Data - Test	TBLE
TS910	GT92002	Data Dictionary Glossary	Data Dictionary	GT
TS910	GT9860A	Object Librarian	Object Librarian	GT
TS910	GT9862A	Business Function Notes	Object Librarian	GT
TS910	GT98DSA	Data Structure Notes-Structure	Object Librarian	GT
TS910	GT98DSB	Data Structure Notes-Structure and Item	Object Librarian	GT
TS910	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.



Code

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

Description

A user defined name or remark.

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 help you track your object mappings.

Some of these reports are called when you select that report from the 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.

Batch Versions - [Work With Batch Versions - Available Versions]

File Edit Preferences Form Row Window Help

Select Find Add Copy Del... Close Seg... New... Dis... Abo Links Availa... OLE ... Internet

Batch Application: R9861101 Job Control Status Master Calling UBE

Read Only Report (Y/N): N

Version	Version Title	User	Last Modified	Security	Description	Client Platform
XJDE0001	Job Control Status Master Driver UBE	DEMO	1/23/2002	0	No Security	

Find records Row:1

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 user 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 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

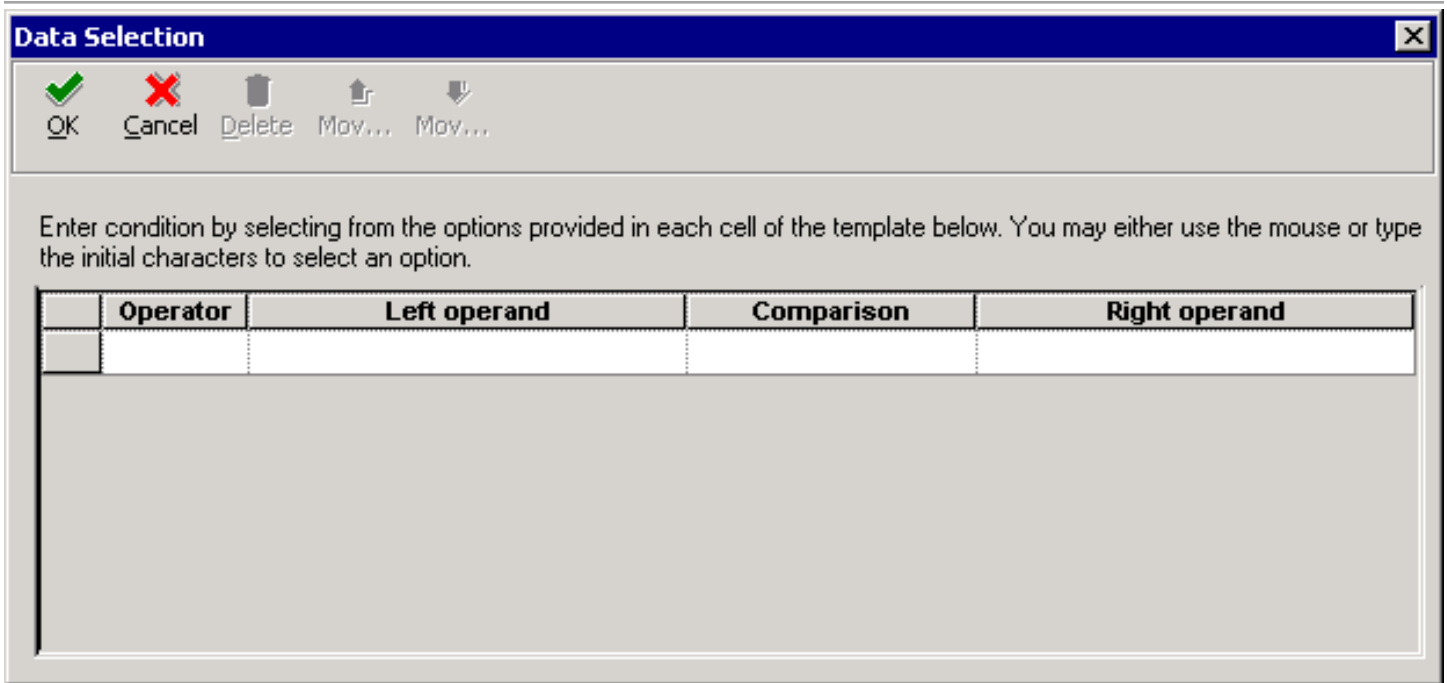
XJDE0002 = Jobs With Done Status

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.



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 Object Configuration 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:

- o Environment Name
- o Object Name
- o Primary Data Source
- o User
- o Object Type
- o Data Source Mode
- o Secondary Data Source
- o 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.

7 Understanding Application Communication

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.

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	<p>Provides platform-independent application program interfaces (APIs) for multiple database access. These APIs are used in these two ways:</p> <ul style="list-style-type: none"> • By 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. <p>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.</p>

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.

Middleware	Protocol Layers
	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 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 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 EnterpriseOne loaded, only one 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 EnterpriseOne server.

Servers also have a net process called JDENet. This process communicates with the client workstations and routes request messages to appropriate dedicated 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 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 EnterpriseOne network process, which is referred to as a JDENET_n job. This job handles network connections and traffic for 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 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 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 *IBM i* 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 for i* handle the same SQL statement:

DBMS	SQL Statement
Microsoft SQL Server	SELECT * FROM PRODDTA.F0101
Oracle	SELECT * FROM PRODDTA.F0101;
<i>DB2 for i</i>	SELECT * FROM PRODDTA/F0101

The purpose of a database middleware layer is to provide a common interface to interpret the various versions of SQL. 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 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 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.

Note:

- *Adding an Environment.*
- *Understanding Object Configuration Manager.*

Setting Up Object Mapping for Direct-Connect Environments

This section lists the prerequisites and discusses how to set up a master data administration environment.

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.

8 Using the Scheduler Application

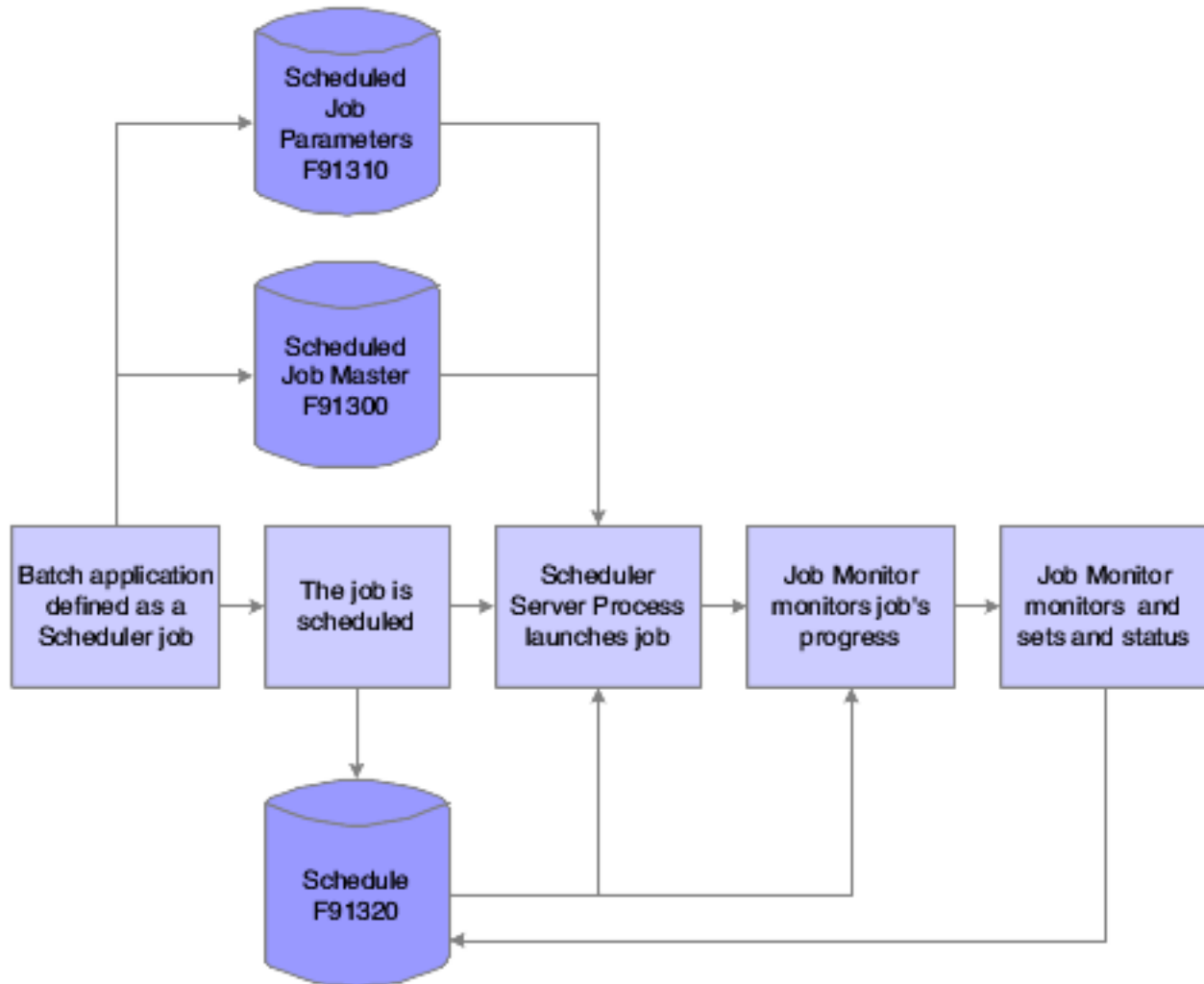
Understanding the Scheduler Application

Occasionally, you might want to run batch jobs that take up a great deal of machine resources or that require users to be signed off after normal working hours. You also might want the flexibility to run jobs at scheduled intervals during the day or even periodically throughout the month or year.

The Scheduler application enables you to schedule batch jobs to run after hours or periodically throughout the day, according to a schedule that you define. You can schedule jobs by time (daily, weekly, monthly, yearly) or based on a specified period. You also can set up the scheduler to restart a job in the event of a job failure.

You can specify the server on which you want the job to run, as well as the time zone, regardless of the locale. The Scheduler system uses a modified version of Universal Coordinated Time (UCT), which counts the number of minutes, not seconds.

The following graphic illustrates the scheduling process:



When you define a scheduled job, the parameters of that job are stored in the Scheduled Job Master table (F91300).

After the job is scheduled, the system writes records to the Schedule table (F91320), indicating each time that the job should be launched. As the job runs, the Job Monitor monitors the progress of the job.

When the job ends, the Job Monitor assigns an end status to the job and updates the record of the job in the Job Schedule table to indicate that the job either ended successfully or in an error.

Working with the Job Scheduler

This section provides an overview of the Job Scheduler in EnterpriseOne and discusses how to:

- Set processing options for the Job Scheduler.
- Schedule a job.
- Schedule a recurring job.
- Revise a scheduled job.
- Review all jobs or local jobs.

Understanding the Job Scheduler

When you schedule a batch process to run through the Scheduler, you can also add a recurrence pattern to the job, which means that the job will restart at the intervals that you define, such as once a week, once a month, or once a year. You can also specify how many times you want the job to run before it ends, or you can define a date after which the job will no longer run.

You schedule jobs in the local time of the server on which the job will run. For example, when you schedule a job, you might select the version that specifies the Eastern Time zone to run jobs in Eastern Standard Time (EST).

Enhanced Scheduler Password Encryption (Release 9.2.7.3)

With Tools Release 9.2.7.3, JD Edwards EnterpriseOne uses enhanced encryption to store the scheduled job user's password in the Scheduled Job Master table (F91300). When a new scheduler record is created, the Schedule Jobs application (P93100) does not display the encrypted password in the Scheduling Advanced Options form (W91300I).

Note: If new records are created using a release prior to Tools Release 9.2.7.3, the system creates scheduler records without enhanced password encryption.

Prerequisites to create and validate records with enhanced password encryption:

- The EnterpriseOne Tools Release 9.2.7.3 or later is deployed on all Windows Clients and Enterprise Servers, updating the F91300 table.
- The Scheduler server which validates the encrypted password is running on EnterpriseOne Tools Release 9.2.7.3 or later.

Note: If the Scheduler kernel (Scheduler server) that validates the enhanced password encryption is running on a release prior to Tools Release 9.2.7.3, the validation fails and the scheduled job is not launched.

You might encounter an issue if some machines are updated and some are not. For example, if the machine that creates scheduler records is updated (creates new records with enhanced encryption) but the Scheduler server is not updated, then the newly created scheduler records are not processed. The Scheduler server, which is not updated, cannot read the records with enhanced password encryption. Similarly, a scheduler record created using the new format cannot be modified by a machine that is not updated to Tools Release 9.2.7.3 or later.

Prerequisite

To use a server's time zone, you must first specify the time zones that you want to use. To do this, copy the Scheduler processing options (version ZJDE0001 on the Work with Versions form), and modify them according to your needs. If you use more than one time zone, you should modify the processing options to display the Work with Versions form each time that you invoke the Schedule Jobs application. That way, you can select the correct version for the time zone in which you want to schedule the job.

Note:

- *Scheduling a Recurring Job.*
- *Setting Processing Options for the Job Scheduler.*

Forms Used to Work with the Job Scheduler

Form Name	FormID	Navigation	Usage
Work With Versions	W983050B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300)	Locate the version that specifies the time zone in which the scheduled job will run.
Work With Scheduled Jobs	W91300B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300). On the Work With Versions form, click the version and then click Add.	Access forms to schedule a job.
Scheduling Information Add	W91300A	On the Work With Scheduled Jobs form, click Add.	Schedule a job.
Scheduling Advanced Options	W91300I	On the Scheduling Information Add form, select Advanced Options from the Form menu.	Enter the user and machine information for the scheduled job.
Recurring Scheduling Information Revisions	W91300C	On the Scheduling Information Add form, select Recurrence from the Form menu.	Schedule a recurring a job.

Setting Processing Options for the Job Scheduler

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

Display

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

View Local Time

Local Time Zone

Enter '1' to adjust time for daylight savings. Enter '0' to never adjust for daylight savings.

Use this Daylight Savings Rule when adjusting for daylight savings.

Process

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

Maximum number of job schedule records

Defaults

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

Default Job Type

Number Of Job Occurrences

Max Number of Job re-submissions

Scheduling a Job

Access the Work With Scheduled Jobs form. On the Work with Versions form, double-click the version that specifies the time zone in which the scheduled job will run.

Note: If you use only one time zone, you might not be prompted to select a version. In this case, the Work With Versions form does not appear. You can delete the Work With Versions form in the menu properties for P91300. By default, GH9015/P91300 prompts for the version.

1. On the Work With Scheduled Jobs form, click Add.
2. On the Scheduling Information Add form, in the Scheduled Job Name field, enter a name that uniquely identifies to the system and the user of a scheduled job.

Use this name to indicate the job function, such as **Monthly Close** or **Nightly Back Up**.

3. In the Scheduled Job Status, determine the status of the scheduled job.

As long as the status is active, the Scheduler determines if the job should be submitted to the server for execution. When the scheduled end date for the job has been reached, the status changes to **Not Active**. To stop the Scheduler from considering the job for submission, you can change the status to **Not Active** (or suspended) at any time prior to the end date. You can reactivate the job if you want the Scheduler to include the job again, but you can reactivate a job only if the end date is in the future.

4. In the Scheduled Batch Application, specify the object name of the report that the Scheduler submits to the server.
5. In the Scheduled Version, specify the version of the report that is scheduled to run. A version identifies a specific set of data selections and sequencing settings that the batch job uses.
6. In the Scheduled Start Date/Time, determine the next date on which the Scheduler submits the scheduled job to the server for execution, and then click OK.

Scheduling a Recurring Job

Access the Recurring Scheduling Information Revisions form.

1. Select one of these options, and complete the accompanying fields that appear after you select an option:

- By Time

Run the job every n days or every weekday.

Run the job at the specified time interval. For example, run the job every 40 minutes or every eight hours.

- Daily

Run the job at the specified interval of days or every weekday. For example, run the job every seven days or every weekday.

- Weekly

Recur every n weeks on Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.

Run the job at the specified weekly interval on the specified day of the week. For example, run the job every two weeks on Monday.

- Monthly

Day n of every x months or the [first, last, and so on] [day, Sunday, Friday, and so on] of every n months.

Run the job on the specified day of every n months or on a specific day of every n months. For example, run the job on day one of every four months or on the last Friday of every second month.

- Period

Day n of every x periods or the [first, last, and so on] [day, Sunday, Friday, and so on] of every n periods.

- Company

- Yearly

Every [month name] [date] OR

The [first, last, and so on] [day, Sunday, Friday, and so on] of [month name] OR

Day x of the [first, last, and so on] period OR

The [first, last, and so on] [day, Sunday, Friday, and so on] of the [first, last, and so on] period.

Schedule the job at the specified time of the year. For example, you might want to run the job on the last day of December of each year, every January 1, on the first Monday of June, on day 15 of the last period, or on the second Tuesday of the fourth period.

2. Specify when you want the Scheduler to stop submitting the job by selecting one of these options:

- No end date
- End after
- End by

Enter the month, day, and year on which you want the job to expire.

3. Click OK.

Revising a Scheduled Job

Access the Work With Versions form.

You might want to revise the information for a job. For example, you might want to change the job status to **Active** or **Not Active**, enter a new batch process as the scheduled job, or change the job start date and time.

1. On the Work With Versions form, select the version that specifies the time zone in which the job will run and click Select.
2. On the Work with Scheduled Jobs form, select the job that you want to revise, and then select Job Revisions from the Row menu.
3. On the Scheduling Information Revisions form, modify the following fields as necessary and click OK:
 - o Scheduled Job Status
 - o Scheduled Batch Applications
 - o Scheduled Version
 - o Scheduled Start Date/Time
4. If you want to remove recurrence from a scheduled job, select Remove Recurrence from the Form menu.
5. If you want to add category codes to the scheduled job, select Category Codes from the Form menu.
6. On the Scheduler Category Codes form, if you want to revise the advanced options for this job, select Advanced Options from the Form menu.

The Scheduling Advanced Options form appears.

Note: You can also activate or inactivate a job by selecting the job on the Work With Scheduled Jobs form, and then selecting Change Status from the Row menu.

Reviewing All Jobs or Local Jobs

Access the Work With Scheduled Jobs form.

If necessary, you can review all of the jobs in all time zones, or local jobs only. Depending on the view that you are currently using, the system protects the other choice. For example, if you are currently viewing local jobs, the system protects the Local Jobs choice and enables you to select only All Jobs.

1. On the Work With Scheduled Jobs form, select Display from the Form menu.
2. Click either All Jobs or Local Jobs.

Using Advanced Scheduling Options

This section provides an overview of the advanced scheduling options and discusses how to:

- Override the environment.
- Override job expiration specifications.

- Define when the Scheduler resubmits a job.
- Override batch application specifications.
- Add values to a report interconnect.

Understanding Advanced Scheduling Options

You can use advanced options to override the job properties, such as the location where the job will run and the environment in which it will run. You can also use advanced options to specify whether you want the system to resubmit a job if it ends in error or if you want to change job expiration specifications.

You can define whether to log errors to the jde.log or jdedebug.log, and whether you want the system to override printer locations and job queues. You can also add a report interconnect to a job if you want to pass parameters to it.

Note: To restore the default values in the advanced options application, click the default button on the Scheduling Advanced Options forms.

Forms Used to Work with Advanced Scheduling Options

Form Name	FormID	Navigation	Usage
Work With Scheduled Jobs	W91300B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300)System. On the Work With Versions form, click the version and then click Add.	Access forms to schedule a job.
Scheduling Advanced Options	W91300I	On the Work With Scheduled Jobs form, select the job, and then select Advanced Options from the Row menu.	Override the environment, job expiration, and batch application specifications. Define when the Scheduler resubmits a job.

Overriding the Environment

Access the Scheduling Advanced Options form. Select the Launch Overrides tab.

A situation might exist in which you need to override an environment. For example, you would override the environment if the environment in which you want to run the job is not available or is different from the environment that you were logged into when you scheduled the job.

Overriding Job Expiration Specifications

Access the Scheduling Advanced Options form. Select the Job Expirations tab.

Job expiration specifications ensure that servers do not become overloaded with unexpired jobs. If necessary, you can override job expiration specifications so that the job never expires, or expires after a certain number of minutes.

For example, suppose you schedule a job to run at midnight and another job for 1:00 a.m., but the server goes down and probably will not come back up again before the jobs are scheduled to run. In this case, you can specify that the first job, which you scheduled to run at midnight, expires in 30 minutes (12:30 a.m.), so that if the server does not come back up within 30 minutes, the job expires.

Defining When the Scheduler Resubmits a Job

Access the Scheduling Advanced Options form.

You can define when the Scheduler resubmits jobs. This feature is useful if a job ends in error, for example, because the Scheduler will submit the job after a certain period of time.

To avoid the use of system resources, you can limit the number of times that a job can be resubmitted. You also can have the Scheduler check for connection errors or runtime errors when the job runs. Connection errors occur when the system fails to connect to the server to submit the job. Runtime errors occur when the server on which the job is running places the job in an error state. You can set up the system to monitor for both cases.

To define when the system resubmits jobs:

1. On the Scheduling Advanced Options form, select the Job Resubmission tab.
2. Specify the number of minutes that elapse before the job continues or terminates, and then activate one of the following options:
 - o Let the job continue.
 - o Terminate the job.
 - o Terminate the job and resubmit.
3. If you want to terminate the job and resubmit it after a certain period of time, select the terminate the job and resubmit after option, and enter the number of minutes that you want to elapse before the system resubmits the job.
4. Select one of the following options that apply when the job ends in error:
 - o Do nothing.
 - o Resubmit immediately.
 - o Resubmit after.
5. Specify whether you want the system to check for connection errors, runtime errors, or both.
6. In the Max Number of Job Resubmissions field, specify the maximum number of times that you want the job to be resubmitted.
7. Click OK.

Overriding Batch Application Specifications

Access the Scheduling Advanced Options form.

You can specify if you want errors written to the jde.log. If you want errors written to the jdedebug.log, you can set the trace level to determine what types of errors to include in the log.

You can also override the printer at which a report is printed. This feature is useful if a specific printer is down or if you want to print a report to a printer other than the default. You can specify whether a job should be printed immediately upon completion or held in the job queue to be printed later.

Also, you can override the queue to which the output of a submitted job is sent. If you want to pass parameters to a particular batch job, you can attach a report interconnect through Batch Application Specifications.

To override batch application specifications:

1. On the Scheduling Advanced Options form, select the Batch Application Overrides tab.
2. Select one or more of the following options:
 - o Jde.log
 - o Jdedebug.log

If you select jdedebug.log, you must also select the jde.log option.
3. Complete these fields:
 - o UBE Logging Level

If you select jdedebug.log, you can set a trace level to log certain levels of errors.

 - o Printer Name

Enter the name of the printer to which you want to print the report that the job generates. If you want to use a default printer, specify **DEFAULT**.

 - o Print Immediate

If you want the job output to be sent to the printer immediately, select this option.

 - o Job Queue

Enter the name of the job queue to which you want the job output sent.
4. Click OK.

Adding Values to a Report Interconnect

Access Work With Scheduled Jobs form.

You can add values to be passed through a report interconnect into a batch process when that batch process is launched. The batch process must first contain a report interconnect.

1. On the Work With Scheduled Jobs form, select the job and then select Advanced Options from the Row menu.
2. On the Scheduling Advanced Options form, select Parameters from the Form menu.
3. On the Report Interconnect form, enter the values that you want to pass to the batch process when the process runs.

4. Click OK.

Reviewing the Job Schedule

This section provides an overview of how to review scheduled job information and discusses how to:

- Review all job schedules.
- Change the launch status of a job.
- View job details.
- Set the job status manually.
- Reset the job schedule.

Understanding Job Schedule Reviewing

When you schedule a job that includes a recurrence pattern, the system creates a set of schedule records, or instances, for the job in the F91320 table. The F91320 table indicates the times and dates that the job will run. You can review these instances and their statuses, and also change the scheduled job information. For example, you can change the location at which you want a job to process, delete a job instance, or override any advanced functions.

Note: Because the F91320 table is also used for audit information, you can modify or delete only jobs that have not yet run.

Note:

- *Revising a Scheduled Job.*

Forms Used to Review the Job Schedule

Form Name	FormID	Navigation	Usage
Work With Versions	W983050B	Job Scheduler (GH9015), Schedule Jobs.	Locate a version that you want to review.
Work With Scheduled Jobs	W91300B	On the Work With Versions form, select a version and then click Select.	Review all scheduled jobs for a version.
Work With All Job Schedule	W91300M	On the Work With Scheduled Jobs form, select All Schedules from the Form menu.	Review all scheduled jobs for a version.

Form Name	FormID	Navigation	Usage
Job Schedule	W91300F	On the Work With Scheduled Jobs form, select the job, and then select Job Schedule from the Row menu.	Change the launch status of a job instance. Reset the job schedule.
Job Maintenance	W986110BC	On the Job Schedule form, select the job, and then select View Detail from the Row menu.	View details about a job, as well as the job queue, the priority in which the job will run, and the location of the report printer. From this form, you can also change the job priority or the location where the report will print.
Manually Set Job Status	W91300L	On the Job Schedule form, select the job instance for which you want to manually set the job status, and then select Set Status from the Row menu.	Set the job status manually.

Reviewing All Job Schedules

Access the Work With All Job Schedule form.

When you review all of the job schedules, you can view all instances of jobs that have been launched. You can even revise a job by choosing a job instance and then selecting Revise Job from the Row menu.

You can filter the job instances that you want to review by launch date, start date, and time. For example, you can review all of the job instances for today's date by entering that date in the Scheduled Start Date field. Or you can review all job instances that were launched on a certain date by entering that date in the Job Launch Date field. You can also filter job instances by scheduled job name, launch status, report name, or scheduled version.

1. On the Work With All Job Schedule form, filter by start date or start time by completing these fields:
 - o Scheduled Start Date
 - o Scheduled Start Time
2. To filter by launch date or launch time, complete these fields:
 - o Job Launch Date
 - o Job Launch Time
3. To filter by job name, launch status, report name, or scheduled version, complete these fields:
 - o Scheduled Job Name
 - o Launch Status
 - o Report Name
 - o Scheduled Version

4. To view all scheduled jobs in all time zones, select All Time Zones from the Form menu.
5. To view all scheduled jobs in the local time zone, select Local Time Zone from the Form menu.

Changing the Launch Status of a Job

Access the Work With Version form.

You can change the launch status of a job. For example, you might need to put a job on hold or reschedule a job.

1. On the Work With Versions form, select the time zone in which the job will run, and then click Select.
2. On the Work With Scheduled Jobs form, locate the job that you want to change.
3. From the Row menu, select Job Schedule.
4. On the Job Schedule form, select the job instance for which you want to change the status, and then enter a new status in the Launch Status field.

Enter **1** for a scheduled status or **50** for hold.

5. Click OK.

Viewing Job Details

Access the Job Schedule form.

1. On the Job Schedule form, select the job, and then select View Detail from the Row menu.
2. On the Job Maintenance form, complete the following fields, if necessary, and click OK:
 - o Job Priority
 - o Printer Name

Setting the Job Status Manually

Access the Work With Scheduled Jobs form.

As a system administrator, you can change the status of jobs if the Scheduler is not updating the launch status or if the Job Monitor is disabled.

Note: If you need to stop a job, select Work with Servers from the Form menu on the Work with Scheduled Jobs form.

Note: You should secure other users from accessing the Set Status option. Only the EnterpriseOne administrator should have access to this option.

1. On the Work With Scheduled Jobs form, select the job, and then select Job Schedule from the Row menu.
2. On the Job Schedule form, select the job instance for which you want to manually set the job status, and then select Set Status from the Row menu.
3. On the Manually Set Job Status form, complete the Scheduled Launch Status field.

Resetting the Job Schedule

Access the Job Schedule form.

If you configure a job schedule and then change your mind, you can remove the changes and regenerate the job schedule by using the previously defined recurrence pattern. The job schedule is reset to the way it was before you made the changes.

1. On the Job Schedule form, select Reset Schedule from the Form menu.
The system displays the following warning message: *This will remove any custom changes to this job's schedule and regenerate the schedule using the recurrence pattern. Are you sure you want to continue?*
2. Click Yes to confirm resetting the job's schedule.

Working with the Scheduler Server

This section provides an overview of the Scheduler server and discusses how to:

- Stop or restart the Scheduler server.
- Pause the job launcher or job monitor.
- Reset the Scheduler server.
- Refresh the Scheduler server settings.
- Modify the Scheduler server and monitor sleep time.

Understanding the Scheduler Server

The Scheduler server is a process that performs two distinct functions: it launches all jobs at the scheduled times, and it monitors each job's progress and ending state. These functions are started by a JDENET message, as defined in the following kernel type in the jde.ini file:

```
[JDENET_KERNEL_DEF10]
dispatchDLLName=jdekrnl.dll
dispatchDLLFunction=_JDEK_DispatchScheduler@24
maxNumberOfProcesses=1
beginningMsgTypeRange=2001
endingMsgTypeRange=2256
newProcessThresholdRequests=0
numberOfAutoStartProcesses=1
```

The Scheduler launches batch processes in a server, environment, and user combination, based on the information in the F91300 table. After the Scheduler is started, JDENET keeps it in a wait state by calling the Scheduler dispatch function every minute with an idle message. This idle message enables the Scheduler process to check whether it should launch a job or monitor the jobs that are running. In addition, JDENET sends the Scheduler any message sent from the workstation (for example, messages that new job schedules were added).

You can stop, reset, restart, and refresh the Scheduler server. For example, if the server goes offline, it needs to be reset. You can also modify the server and monitor sleep time, specifying how many seconds you want JDENET to wait until it checks to see if it needs to initialize, or wake up, the Scheduler server.

You also might encounter situations for which you need to activate or deactivate the Job Launcher or Job Monitor. For example, you might need to take down the servers to which you submit jobs and for which you want to avoid unnecessary connection errors when jobs are submitted.

You can also change the `jde.ini` file to enable the Scheduler to restart automatically by changing the `numberOfAutoStartProcesses` line. If you enable this feature, and the server on which the Scheduler server is running comes down, the Scheduler server automatically restarts when the server comes back up, instead of having to be restarted manually. When the Scheduler server restarts, the Scheduler checks the F91320 table to determine if it should restart on that server. If not, the Scheduler shuts down.

JDENET handles the calls to initialize the Scheduler server. The JDENET process either sends a message to initialize the Scheduler to launch a job if it receives a message that table F91320 has changed, or it sends an idle message if no change is detected. For faster response time, you can decrease the number of seconds that you want JDENET to wait until it checks to see if table F91320 has changed.

Note: This application is for administrators only. You should secure users from accessing the Scheduler server application.

Control Record

A control record is a job record in the F91300 table. It is named *SCHEDULE and is hidden from the user. The *SCHEDULE record contains information about the state of the Scheduler processes on the server, and it is the method of communicating to those processes.

For example, when the launch loop starts on the server, it sets a flag in this record to indicate that it is up and running. You can end the launch loop by toggling the corresponding end process flag (such as Job Launcher Status or Job Monitor status) from the Form menu on Scheduler Server Control. The next time that the launch loop fetches the control record, it finds the flag, resets both flags, and ends.

If the system does not find the control record when it is fetched, the record is recreated by P91300 when entering the Scheduler Server Control form. In addition, if the record is corrupt, the aforementioned function is called to recreate it. The sleep times for the job monitor are reset to 15 minutes, and the audit information in this record is updated with the user ID set to SCHEDULER.

Dispatch Function

The dispatch function handles the incoming message from the workstation and starts the requested process. The JDENET process either sends a message to initialize the Scheduler, signals that the F91320 table has changed, or gives an idle message. The idle message is sent every minute unless one of the other messages is sent. When the idle message is sent, the dispatch function checks to see if the launch loop or job monitor needs to be called. If neither does, control is given back to JDENET.

Launch Loop

The launch loop function selects all of the jobs up to the current time. It then loops through the selected records and launches the active jobs if they have not expired. After launching all current jobs, the launch loop fetches all future jobs sorted by start time. If the fetch succeeds, the next select time (NST) is set to the difference between the current time and the start of the next job. If the fetch fails, the NST is set to zero, which indicates that this function should be run the next time that any record is added to or updated by the F91320 table. In addition to launching jobs, the launch loop checks the control record periodically to see if it should exit.

The launch loop also looks for updates of all the schedule instances (F91320 records) and job headers (F91300 records) that it fetches. After the launch loop has processed these records, it then commits any changes and unlocks all of the records.

Job Monitor Loop

The job monitor loop monitors the ending statuses of the launched jobs and relaunches those that end in error if requested to do so by the user. This loop cycles through the internal job list that the job launch loop populates. In addition, it terminates jobs that run too long, if requested to do so. A job cannot be relaunched for more times than specified in the F91300 record of the job.

Like the launch loop, the job monitor loop periodically fetches the control record to see if it should end.

Note:

- *"Defining Report Interconnects" in the JD Edwards EnterpriseOne Tools Report Design Aid Guide .*
- *"Using Debug Tracing" in the JD Edwards EnterpriseOne Tools APIs and Business Functions Guide .*

Forms Used to Work with the Scheduler Server

Form Name	FormID	Navigation	Usage
Work with Versions	W983050B	Job Scheduler (GH9015), Schedule Jobs (P91300).	Locate the version that specifies the time zone in which the scheduled jobs run.
Work with Scheduled Jobs	W91300B	On the Work with Versions form, select a version and then click Select.	Access the Scheduler Server Control form.
Scheduler Server Control	W91300G	On the Work with Scheduled Jobs form, select Scheduler Server from the Form menu.	Stop, restart, reset, refresh, or modify the Scheduler server. Pause the job launcher and job monitor. Modify the Job Monitor sleep time.

Stopping or Restarting the Scheduler Server

Access the Work with Versions form.

1. On the Work with Versions form, select the version that specifies the time zone in which the scheduled jobs run, and then click Select.
2. On the Work with Scheduled Jobs form, select Scheduler Server from the Form menu.
3. On the Scheduler Server Control form, perform one of these operations and click OK:
 - o To stop the server, select Stop Scheduler from the Form menu.

- To restart the server, select Start Scheduler from the Form menu.

Pausing the Job Launcher or Job Monitor

Access the Scheduler Server Control form.

Note: You might want to pause the job launcher or job monitor, such as when you want to take down the servers to which you submit jobs, and you want to avoid server connection errors that might occur while those servers are down. When you pause the job launcher, the Scheduler stops looking at the F91320 table for jobs to launch. When you pause the job monitor, the Scheduler stops monitoring the status of launched jobs.

1. On the Scheduler Server Control form, select Pause Job Launcher from the Form menu to pause the job launcher.
2. To pause the job monitor, select Pause Job Monitor from the Form menu.

Resetting the Scheduler Server

Access the Scheduler Server Control form.

Note: You reset the Scheduler server after you change the status of the Job Monitor or Job Launcher. For example, if you change the status of the Job Monitor, you would select Reset to refresh the settings on the server.

1. On the Scheduler Server Control form, select Reset from the Form menu.
2. Click OK.

Refreshing the Scheduler Server Settings

Access the Scheduler Server Control form.

Note: When you refresh the Scheduler server settings, the server refreshes its cache of launched jobs, and closes and restarts all environment and table handles. It is a refresh of the server's internal structures. You might want to refresh the Scheduler server settings if you had to restart the server.

1. On the Scheduler Server Control form, select Refresh from the Form menu.
2. Click OK.

Modifying the Scheduler Server and Monitor Sleep Time

Access the Scheduler Server Control form.

Note: Sleep time is the time that the Scheduler server or monitor is idle.

Scheduler Sleep Time

This field indicates the number of seconds that the scheduler server will sleep (or idle). For example, if this field is set to 60 seconds, the Scheduler server checks every 60 seconds to see if it needs to launch or monitor jobs. The default is 60, and it must be greater than zero.

Job Monitor Sleep Time

This field indicates the number of minutes the job monitor will pause between job status checks.

Creating Daylight Savings Rules

This section provides an overview of daylight savings rules and discusses how to:

- Add daylight savings rules.
- Add an additional daylight savings record to a daylight savings rule.
- Revise daylight savings rules.

Understanding Daylight Savings Rules

Daylight savings rules tell the system how each locale implements its daylight savings time. You associate a daylight savings rule to each user profile depending on the geographic location of the user. In addition, you can assign a daylight savings rule to a batch job that is scheduled to run in the Scheduler. The Scheduler uses these daylight savings rules, along with time zone information, to determine when batch jobs run on a particular server.

You can only associate one daylight savings rule with a user or scheduled job. However, in the Daylight Savings Time program (P00085), you can create multiple records for the same rule to account for variations in the daylight savings time from one year to the next. For example, if the daylight savings time period for a region is the same from 2008 through 2010, and then changes for 2011, you can create one daylight savings time rule that contains the following two records:

- DST Rule1–2008–*Start and end dates for the daylight savings period*
- DST Rule1–2011–*Start and end dates for the daylight savings period*

When the DST Rule1 in this example is associated to a user or scheduled job, EnterpriseOne uses the 2008 daylight savings record to determine the time until 2011, when it will automatically use the daylight savings time dates defined in the rule for 2011.

In the Southern Hemisphere, daylight saving time starts in one year and ends in another. When you set up a daylight savings time rule, the year that you enter for the daylight savings time rule must be the year in which the daylight savings time begins.

Prerequisites

The daylight savings time functionality that enables you to create multiple records for one rule is an EnterpriseOne Tools feature that is applicable to the EnterpriseOne Applications 8.12 and 9.0 releases. This feature comes automatically with the 9.0 release. However, to obtain this feature for the 8.12 release, you must download an electronic software update (ESU). On My Oracle Support, access SAR 8414822 to locate the ESU. In addition to the Daylight Savings Rule program changes, the ESU contains instructions on performing a table conversion to convert existing daylight savings rule data from the F00085 table to the F00085A.

Adding Daylight Savings Rules

From the Job Scheduler menu (GH9015), select Daylight Savings Rules (P00085).

1. On Work With Daylight Savings Rules form, click Add.
2. On the Add Daylight Savings Rule form, complete these fields:
 - o Rule Name
Enter a unique name for the daylight savings rule.
 - o Daylight Savings Effective Year
Enter the year in which the daylight savings rule begins.
 - o Description
Enter a description of the daylight savings rule.
3. In the Rule Type area, select one of these options to indicate the method that you want to use to determine the daylight savings rule dates:
 - o By Day of Week Instance
Select this option to enter start and stop dates for daylight savings on a certain day of the week for a certain month, such as the first Sunday of April to the first Sunday of October.
 - o By Day of the Month
Select this option to enter start and stop dates for daylight savings on a certain day of a certain month, such as April 3 to October 10.
4. Depending on the method that you chose, complete the remaining fields to specify when daylight savings starts and ends, and then click OK.

Adding an Additional Daylight Savings Record to a Daylight Savings Rule

To add another set of dates to accommodate for a change in the daylight savings period, you can add another record that contains new dates to an existing daylight savings rule.

From the Job Scheduler menu (GH9015), select Daylight Savings Rules (P00085).

1. On the Work With Daylight Savings Rules form, click Copy.
2. On the Add Daylight Savings Rule form, complete these fields:

Note: Do not modify the Rule Name field because doing so will create a new rule that you will have to reassign to a user profile or scheduled job.

- o Daylight Savings Effective Year.
Enter the year in which the daylight savings rule begins.
- o Description.

3. In the Rule Type area, select one of these options to determine which method that you want to use to determine the daylight savings rule dates:
 - o By Day of Week Instance.
 - o By Day of Month.
4. Depending on the method that you chose, complete the remaining fields to specify when daylight savings starts and ends, and then click OK.

Revising Daylight Savings Rules

When you revise an existing daylight saving time rule, you cannot change the rule name or the year that the daylight savings rule takes affect.

From the Job Scheduler menu (GH9015), select Daylight Savings Rules (P00085).

1. On the Work With Daylight Savings Rule form, select the rule that you want to revise and then click Select.
2. In the Rule Type area, select the method that you want to use to determine a daylight savings rule:
 - o By Day of Week Instance.
 - o By Day of Month.
3. Depending on the method that you chose, complete the remaining fields to specify when daylight savings starts and ends, and then click OK.

Running Scheduler Reports

Run the Scheduled Jobs report when you want to review a summary of scheduled jobs and their statuses. You can use processing options to specify whether to run this report based on UCT or local time. You also can adjust for daylight savings time.

If you want to purge records from the F91320 table, run the Scheduler Purge program. You can run the purge program in proof mode or final mode. You can launch the Scheduler Purge program by doing a Fastpath to BV, then run Run R91300B (XJDE0001). You can also launch the Scheduler Purge program by doing a Fastpath to menu GH9015, and then launch Purge Scheduled Jobs.

Note:

- *"Batch Versions at Submission" in the JD Edwards EnterpriseOne Tools Report Printing Administration Technologies Guide* .

Printing the Scheduled Jobs or Purge Scheduled Jobs Report

Access the Work With Batch Versions - Available Versions form.

1. Select a version in the detail area, and then click Select.
2. On the Version Prompting form, select one or both of the following options, and then click Submit:
 - o Data Selection

- Data Sequencing
- 3.** On the Report Output Destination form, select one of these options, and then click OK:
- On Screen
 - To Printer

9 Glossary

activity rule

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

CafeOne

An abbreviation for the Composite Application Framework.

charts

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

composite application framework

A user interface framework that enables EnterpriseOne applications to integrate with multiple third-party contents and applications, as well as other EnterpriseOne applications.

edit mode

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

fast path

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

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.

QBE

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

selection

Found on 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.

softcoding

A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.

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 EnterpriseOne form or report.

Index

A

- advantages
 - configurable network computing [3](#)
 - custom solution without consequences [5](#)
 - flexible and leveraged technology [4](#)
 - network-centric software [3](#)
 - worldwide business support [4](#)

B

- batch applications
 - scheduling a jobJob Scheduler [183](#)

C

- communications
 - application [173](#)
 - JDEBase [176](#)
 - JDENet [173](#)
 - kernel processes [176](#)
 - message-based [174](#)
 - network processes [175](#)
 - process-based design [175](#)
 - socket-based [174](#)
- configurations
 - basic environments [11](#)
 - configuration data [22](#)
 - data sources [19](#)
 - environment specific data sources [11](#)
 - recommended [11](#)
 - remote environments [14](#)

D

- data selection
 - Data Source Master Compare Report [60](#)
 - Data Source Master Report [60](#)
 - Job Master Deletion by Days Old Report [161](#)
 - Object Configuration Batch Applications [166](#)
 - Verify Object Configuration Mappings Report [64](#)
- data sources
 - adding
 - database [46](#)
 - logical [46](#)
 - client access names [46](#)
 - configuration
 - creating new data class [119](#)
 - data class [119](#)
 - data load [120](#)
 - table data classes [119](#)
 - templates by environment [119](#)
 - database tables [31](#)
 - definition [45](#)
 - definitions [32](#)
 - environment [118](#)
 - machine name [32](#)
 - names [31](#)
 - naming conventions [46](#)
 - planning [45](#)
 - reports [56](#)

- set up [45](#)
- shared [118](#)
- system connections [44](#)
- table owner [46](#)
- types [31](#)
 - central objects [34](#)
 - control table [34](#)
 - required [32](#)
 - versions [34](#)
- database structures [35](#)
 - DB2 for IBM i Server [40](#)
 - IBM DB2 for LUW (Linux, UNIX, Windows) 8.1.4 [42](#)
 - Oracle [37](#)
 - table spaces [37](#)
 - tables [37](#)
 - user schema [37](#)
 - SQL Server [38](#)
- daylight savings rules
 - adding [197](#)
- dispatch functions, using with the Scheduler server [193](#)

E

- Environment Director [82](#)
 - different environments [118](#)
 - Director Mode [86](#)
 - Express Mode [110](#)
- environments
 - adding [71](#)
 - copying [74, 74](#)
 - creating [86, 110](#)
 - definitions [67, 67](#)
 - deleting [80](#)
 - deployment [14](#)
 - development [13](#)
 - master data administration [178](#)
 - planner [13](#)
 - pristine [13](#)
 - production [12](#)
 - prototype [12](#)
 - set up [69](#)
 - types [67, 67](#)

F

- F91300 table [192](#)
- F91320 table [179, 193, 195](#)
- forms
 - Copy Environments [74](#)
 - Data Load [106](#)
 - Data Source Revisions [47](#)
 - Data Source Selection [71](#)
 - Data Structure Design [120](#)
 - Environment [95, 112](#)
 - Environment Director [94, 111](#)
 - Environment Director Revisions [107, 116](#)
 - Environment Properties [97](#)
 - Environment Revisions [71](#)
 - Host Planner [79](#)
 - Machine Search ((amp)) Select [47, 135, 139](#)
 - machines [103, 114](#)

- Object Configuration Manager *134*
- Object Management Workbench *120*
- Object Mapping Revisions *135*
- Oracle Database Object Sizing *148, 151*
- Path Code *98*
- Path Code Properties *99*
- Path Code Revisions *76*
- Release/Data Source Map Revisions *53*
- Revise Database and Data Source Overrides *148*
- Revise OL Data Source *139*
- Versions Prompting *142*
- Work With Batch Versions - Available Versions *57, 61, 146, 159, 163, 167, 168, 169, 170, 171*
- Work With Bath Versions - Availavle Versions *142*
- Work With Data Sources *47*
- Work With Environments *70, 76, 80*
- Work With Installation Plans *79*
- Work With Object Mappings *135, 139*
- Work With On Track Planning Setup *123*
- Work With Oracle Database Object Sizing *148*
- Work With Release/Data Source Map *53*

fundamentals *5*

- data sources *6*
- environments *5*
- object deployment *9*
- object storage *7*
- path codes *6*

I

- implementation teams *1*
 - development *2*
 - functional *2*
 - system integration *2*
 - technology *2*

J

JDEBase

- database middleware *173*

JDENet

- application layer *174*
- communications *174*
- communications middleware *173*
- network *173*
- transport layer *173*

JDENET process *192, 193*

- job launcher, activate or deactivate *195*

- Job Maintenance form *190, 191*

- job monitor loops, using with the Scheduler server *194*

- job monitor, activate or deactivate *195*

- Job Schedule form *190, 191, 191, 191*

Job Scheduler

- changing launch status *191*
- resetting the job schedule *192*
- scheduling a job *183*
- setting job status manually *191*
- viewing job details *191*

jobs

- printing job reports
 - Purge Scheduled Jobs report *198*
 - Scheduled Jobs report *198*

M

- Manually Set Job Status form *190, 191*

middleware

- JDEBase
 - database *173*
- JDENet
 - communication *173*

O

- object deployment
 - initial installation *9*
 - workstation installation *9*

object mapping

- generic text *155*
- individual objects *129*
- object types
 - default maps *129*

object storage

- central objects *7*
- replicated objects *8*
- serialized objects *8*

OCM mappings

- creating *83*
- default *83*
- default mapping to local *83*
- default mapping to server *83*
- UBEs and business functions *83*

P

- path codes *25*
 - adding to Enterprise Server *80*
 - at development *25*
 - at runtime *25*
 - central objects *29*
 - definitions *26*
 - installation *25*
 - object storage *26*
 - set up *29*
 - usage *25*

processing options

- Batch Applications *163*
- Create OCM Records for Business Functions *147*
- Data Source Master Compare Report *59*
- Data Source Master Report *59*
- Database Data Sources *51*
- Environment Master (P0094) *70*
- Job Master Deletion by Days Old Report *160*
- Logical Data Sources *51*
- Object Configuration Manager *134*
- Object Configuration System Table Update *145*
- OCM Category Add/Update/Delete Report *171*
- Release/Data Source Map Revisions *55*
- Verify Object Configuration Mappings Report *63*

programs

- Copy System to Planner (R9698611) *118*
- Data Source Templates By Environments (P98503) *119*
- Deployment Server Installation (P986115) *9*
- Environment Director (P989400) *86*
- Environment Master (P0094) *70*
- Object Configuration Global Update (R986110) *169*
- Object Configuration Manager (P986110) *7, 132, 134*
- Object Management Workbench (P98220) *119, 119*
- On Track Planning Setup (P985033) *122*

Release/Data Source Map (P00948) *52*

Workstation Installation (P986115) *9*

Purge Scheduled Jobs report, printing *198*

R

Recurring Scheduling Information Revisions form *182, 184*

remote environments

DV900 Environment *17*

DV900 Object Mappings *17*

PD900 Environment *15*

PD900 Object Mappings *15*

PS900 Environment *18*

PS900 Object Mappings *18*

PY900 Environment *16*

PY900 Object Mappings *16*

reports

Create OCM Records for Business Functions *145*

Data Source Master *56*

Data Source Master Compare *56*

Job Master Deletion by Days Old Report *158*

Object Configuration Copy *169*

Object Configuration Delete *168*

Object Configuration Global Update *167*

Object Configuration Mapping Comparison *166*

OCM Category Update/Delete *170*

Verify Object Configuration Mappings *61*

S

Schedule table (F91320) *179*

Scheduled Jobs report, printing *198*

Scheduler *198*

Scheduler server

dispatch functions *193*

job monitor loops *194*

launch loops *193*

modifying the sleep time *195*

refreshing the settings *195*

resetting *195*

stopping or restarting *194*

using *192*

Scheduler Server Control form *194, 195*

Scheduling Advanced Options form *182, 186, 186*

Scheduling Information Add form *182, 183*

servers

partitioning logic *129*

T

tables

Data Source Master (F98611) *31, 127*

Data Sources by Environment (F98511) *119*

Environment Detail (F00941) *68, 142*

Job Control Status Master (F986110) *34*

Job Number Master (F986111) *34*

Library List Control (F0093) *67*

Library List Master (F0094) *67*

Library List Master File (F0094) *6, 82*

Object Configuration Master (F986101) *34, 68, 127, 128*

Object Path Master File (F00942) *25, 68*

Processing Option Text (F98306) *34*

Release/Data Source Map Table (F00948) *52*

User Overrides (F98950) *34*

Versions List (F983051) *34*

W

Work With All Job Schedule form *189*

Work With Scheduled Jobs *182*

Work with Scheduled Jobs form *194*

Work With Scheduled Jobs form *183, 186, 189, 191*

