Informatica

Informatica® PowerExchange for JD Edwards EnterpriseOne

10.5.6

User Guide for PowerCenter

Informatica PowerExchange for JD Edwards EnterpriseOne User Guide for PowerCenter
10.5.6
May 2024

Publication Date: 2024-05-29

Table of Contents

Preface

Use the *Informatica® PowerExchange® for JD Edwards EnterpriseOne User Guide* to learn how to read from or write to JD Edwards EnterpriseOne by using PowerCenter Client. Learn to create a connection, develop mappings, and run sessions in an Informatica domain.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Understanding PowerExchange for JD Edwards EnterpriseOne

This chapter includes the following topics:

Understanding PowerExchange for JD Edwards EnterpriseOne Overview

PowerExchange for JD Edwards EnterpriseOne integrates with the JD Edwards EnterpriseOne Enterprise Server to extract data from JD Edwards EnterpriseOne sources and write data to JD Edwards EnterpriseOne targets. The PowerCenter Integration Service uses JDBj APIs to extract data from and write data to JD Edwards EnterpriseOne.

Note: JD Edwards EnterpriseOne was formerly called JD Edwards OneWorld.

Understanding JD Edwards EnterpriseOne

JD Edwards EnterpriseOne is an Enterprise Resource Planning (ERP) application with an integrated toolset. You can configure application suites in JD Edwards EnterpriseOne according to your business needs. The application suites support manufacturing, financial, distribution or logistics, and human resource operations for organizations.

Each JD Edwards EnterpriseOne application suite consists of different systems. For example, the Financial Suite contains systems such as Enhanced Accounts Receivable, Accounts Payable, General Accounting, and Fixed Assets. Each system consists of applications, forms, reports, and database tables that are designed to handle specific business needs.

JD Edwards EnterpriseOne also contains environments such as Production and Pristine. You can connect to any environment to access the application suites.

JD Edwards EnterpriseOne Tables and Views

JD Edwards EnterpriseOne maintains ERP data in tables and views that are created in the underlying database. The tables and views used in the JD Edwards EnterpriseOne system are similar in structure to those defined in a relational database. However, JD Edwards EnterpriseOne maintains additional tables that store metadata about the columns in tables and views, such as primary indexes, precision, and scale.

Tables and views are categorized by system codes. JD Edwards EnterpriseOne provides distinct system codes to the systems in the applications suites present in JD Edwards EnterpriseOne.

To extract data and metadata from tables and views, third-party applications can send a request in XML format to the JD Edwards EnterpriseOne Enterprise Server. The enterprise server sends its response in XML format.

JD Edwards EnterpriseOne Business Functions

A business function is an encapsulated set of business rules and logic that can be reused by multiple applications. Business functions provide a common way to access the JD Edwards EnterpriseOne database. A business function performs a specific task. Master business functions provide the logic and database calls necessary to edit and commit a transaction to the database. Third-party applications can use master business functions for JD Edwards EnterpriseOne functionality, data validation, security, and data integrity.

Handling Bulk Data Movement in JD Edwards EnterpriseOne

PowerExchange for JD Edwards EnterpriseOne provides transformations for loading bulk data in JD Edwards EnterpriseOne targets. These transformations enhance the performance of writing large volumes of data to JD Edwards EnterpriseOne.

Understanding JD Edwards EnterpriseOne Interface and Base Tables

JD Edwards EnterpriseOne applications store business application data in database tables known as base tables. When you write external bulk data to JD Edwards EnterpriseOne, the data load process occurs in two parts:

1. Loads external data into interface tables.
2. Executes the business function that invokes the batch process. The batch process extracts data from interface tables and writes the data to base tables.

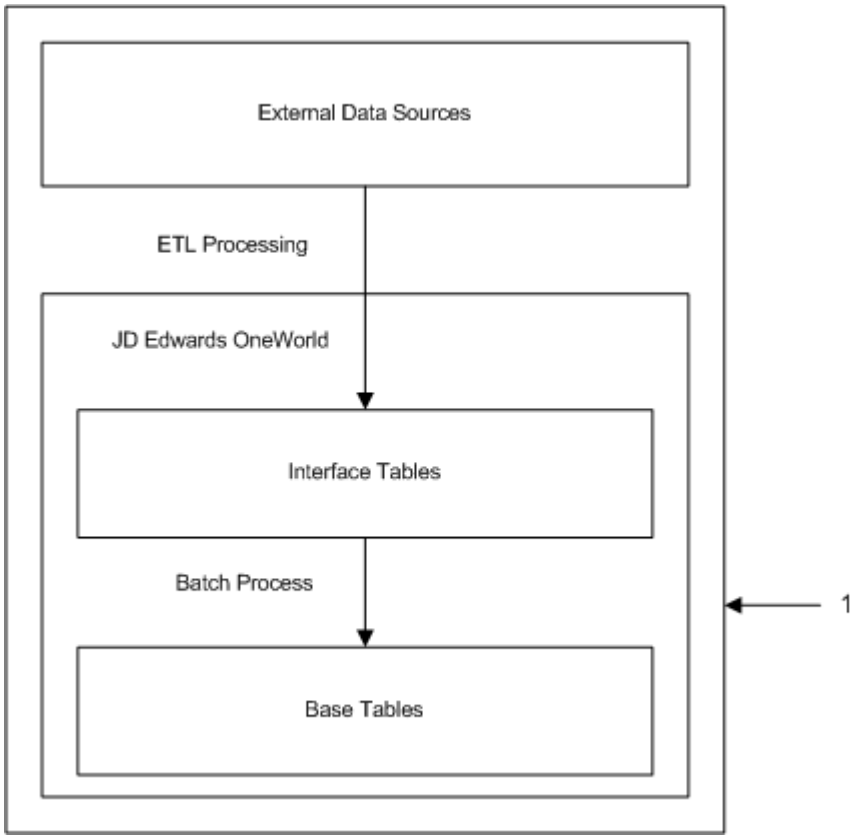
When you load data from the interface table to the JD Edwards EnterpriseOne base table, the batch process associated with interface tables ignores orphan rows present in extension interface tables when a parent row does not exist in the master interface table.

Understanding Transformations for Bulk Loads

You can create the following types of bulk load transformations:

- JDEOne Load transformation. Transforms data from external data sources and loads the data into staging tables, called interface tables or Z tables. JD Edwards EnterpriseOne interface tables mirror JD Edwards EnterpriseOne application tables.
- JDEOne Batch Process Invoker transformation. Connects to JD Edwards EnterpriseOne at run-time and executes the business function that invokes the batch process. The batch process triggers the data flow from an interface table to a JD Edwards EnterpriseOne base table.

The following figure shows the bulk load architecture:



1. The PowerCenter Integration Service invokes a business function that loads the bulk data from the interface table to the base table.

PowerCenter Integration with JD Edwards EnterpriseOne

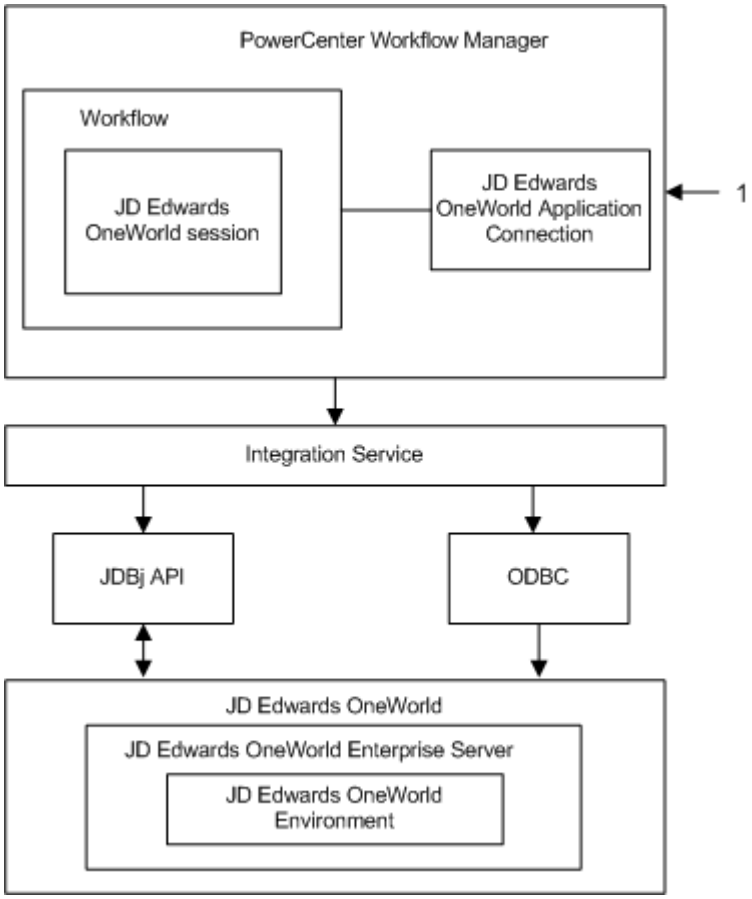
Before you connect to the JD Edwards EnterpriseOne Enterprise Server through the Designer, you must configure the JD Edwards EnterpriseOne Enterprise Server to support interoperability using XML and the JDBj APIs.

You can import JD Edwards EnterpriseOne tables and views as source or target definitions. When you import a source or target definition, the Designer connects to a JD Edwards EnterpriseOne Enterprise Server to import the metadata.

When the Designer connects to JD Edwards EnterpriseOne, it connects to an environment. The Designer sends a request for metadata in XML format to the JD Edwards EnterpriseOne, and then converts the XML response from JD Edwards EnterpriseOne to a source or target definition.

When you run a session, the PowerCenter Integration Service connects to JD Edwards EnterpriseOne to read data from sources and write data to targets. When the PowerCenter Integration Service reads or writes JD Edwards EnterpriseOne data or executes a business function, it connects to the JD Edwards EnterpriseOne Enterprise Server through the JDBj APIs and makes API calls to the JD Edwards EnterpriseOne system.

The following figure shows how the PowerCenter Integration Service integrates with JD Edwards EnterpriseOne:



1. The PowerCenter Integration Service reads and writes data based on session and application connection configuration.

To connect to any environment in the JD Edwards EnterpriseOne system, specify the name of the environment in the JD Edwards EnterpriseOne application connection. The PowerCenter Integration Service reads or writes data after establishing a connection with the specified environment.

CHAPTER 2

PowerExchange for JD Edwards EnterpriseOne Configuration

This chapter includes the following topics:

PowerExchange for JD Edwards EnterpriseOne Configuration Overview

PowerExchange for JD Edwards EnterpriseOne installs with the Informatica services. Before you use PowerExchange for JD Edwards EnterpriseOne, you must complete the configuration tasks.

Configuring PowerExchange for JD Edwards EnterpriseOne

To configure PowerExchange for JD Edwards EnterpriseOne, perform the following steps:

1. Create a registry entry for PowerExchange for JD Edwards EnterpriseOne on the client machine:
   a. Access the following directory:
      `<Informatica installation directory>\clients\PowerCenterClient\client\bin`
   b. Run the `PWX_JDEONEW_64.reg` file to create the registry entry.
2. Register the PowerExchange for JD Edwards EnterpriseOne plug-in.
3. Configure JD Edwards EnterpriseOne to support XML interoperability.
4. Configure JD Edwards EnterpriseOne to support Java API interoperability.
5. Copy JD Edwards EnterpriseOne libraries.

After you configure PowerExchange for JD Edwards EnterpriseOne, you can create connections to access JD Edwards EnterpriseOne. Create connection objects in Workflow Manager so the PowerCenter Integration Service can connect to JD Edwards EnterpriseOne at run time.

Registering the Plug-in

After you create a registry entry for PowerExchange for JD Edwards EnterpriseOne on the client machine, register the plug-in with the repository. If you are upgrading from a previous version, update the plug-in registration when you register the plug-in.

To register the plug-in, the repository must be running in exclusive mode. Use the Administrator tool or the pmrep RegisterPlugin command line program to register the plug-in. If you do not have the correct privileges to register the plug-in, contact the user who manages the PowerCenter Repository Service.

The plug-in file is an .xml file that defines the functionality of the adapter. When you install the server component, the installer copies the plug-in file to the following directory:

```
<Informatica installation directory>/server/bin/plugin
```

The name of the plug-in file for PowerExchange for JD Edwards EnterpriseOne is `JDEOneWorld.xml`.

Registering the Plug-in from the Administrator Tool

Register a repository plug-in to add its functionality to the repository.

1. Run the PowerCenter Repository Service in exclusive mode.
2. In the **Navigator**, select the PowerCenter Repository Service to which you want to add the plug-in.
3. In the **Contents** panel, click the **Plug-ins** view.
4. In the **Actions** menu of the **Domain** tab, select **Register Plug-in**.
5. On the **Register Plugin** page, click the **Browse** button to locate the plug-in file.
6. Enter your user name, password, and security domain.

   The **Security Domain** field appears when the Informatica domain contains an LDAP security domain.
7. Click **OK**.

   The PowerCenter Repository Service registers the plug-in with the repository. The results of the registration operation appear in the activity log.
8. Run the PowerCenter Repository Service in normal mode.

Registering the Plug-in from the Command Line Program

You can use the pmrep RegisterPlugin command to register the plug-in from the command line program.

1. Run the PowerCenter Repository Service in exclusive mode.
2. Run the `pmrep Connect` command to connect to the Repository Service using a user account with Administrator Repository privilege.

   The RegisterPlugin command uses the following syntax:

   ```
   pmrep connect -r <repository name> -d <domain_name> -n <domain user name> -x
   <domain_password>
   ```
3. Find `<adaptername>.xml` in the following directory:

```
<Informatica installation directory>\server\bin\Plugin
```

4. Run the `pmrep RegisterPlugin` command to update the repository.

   The `RegisterPlugin` command uses the following syntax:

   ```
   pmrep registerplugin -i <Informatica installation directory>\server\bin\Plugin
   \<adaptername>.xml -e
   ```

Configuring XML Interoperability

To configure the JD Edwards EnterpriseOne system environment to support XML interoperability, configure the `jas.ini` and `jde.ini` files.

Configuring the jas.ini File

The jas.ini file is located in the JD Edwards EnterpriseOne thin client installation folder. For example, the location of the JD Edwards EnterpriseOne thin client installation folder for Oracle WebLogic is `<WebLogic Installation Directory>\html_server\stage\html_instance\app`.

1. Copy the jas.ini file from the JD Edwards EnterpriseOne thin client installation folder to the following directory:

   ```
   <PowerCenter Installation Directory>\clients\PowerCenter\client\bin\javalib
   ```

2. Open the jas.ini file.
3. Specify or verify values of the following sections in the jas.ini file.

   The following table describes the values of the SECURITY section in the jas.ini file:

Property	Description
`SecurityServer`	Specify the name of the JD Edwards EnterpriseOne security server.

   The following table describes the values of the JDENET section in the jas.ini file:

Property	
`ServiceNameConnect`	Specify the name of the JD Edwards EnterpriseOne Enterprise Server JDENET port.

Configuring the jde.ini File

The jde.ini file is located in the JD Edwards EnterpriseOne Enterprise Server installation directory.

The following table lists the installation directory based on each version of JD Edwards EnterpriseOne:

JD Edwards EnterpriseOne Version	Installation Directory
9.2	<JD Edwards EnterpriseOne 9.2 Installation Directory>\JDEdwards\E920\ini

1. Locate the jde.ini file in the JD Edwards EnterpriseOne Enterprise Server installation directory.
2. Open the jde.ini file.
3. Verify the entries for the LREngine, XML List Kernel, and Call Object Kernel sections in the jde.ini file on the JD Edwards EnterpriseOne Enterprise Server.

   For example, the sample code shows the LREngine, XML List Kernel, and Call Object Kernel sections in the jde.ini file for JD Edwards EnterpriseOne 9.2:

   ```
   [LREngine]
   System=d:\peoplesoft\ddp\e920\output
   Repository_Size=20
   Disk_Monitor=YES
   [JDENET_KERNEL_DEF16]
   krnlName=XML List Kernel
   dispatchDLLName=xmllist.dll
   dispatchDLLFunction=_XMLListDispatch@28
   maxNumberOfProcesses=5
   beginningMsgTypeRange=5257
   endingMsgTypeRange=5512
   newProcessThresholdRequest=0
   numberOfAutoStartProcesses=1
   [JDENET_KERNEL_DEF6]
   krnlName=CALL OBJECT KERNEL
   dispatchDLLName=XMLCallObj.dll
   dispatchDLLFunction=_XMLCallObjectDispatch@28
   maxNumberOfProcesses=10
   numberOfAutoStartProcesses=0
   ```
4. Add entries for the API libraries under CLASSPATH in the JDE_CG section.

   The following table lists the API libraries and their location for each version of JD Edwards EnterpriseOne:

JD Edwards EnterpriseOne Version	API Libraries	Location
9.2	xalan.jar, xerces.jar, and XTS.jar	<JD Edwards EnterpriseOne 9.2 Installation Directory>\JDEdwards\E920\system\classes

   For example, the sample code shows the JDE_CG section for JD Edwards EnterpriseOne 9.2:

   ```
   [JDE_CG]
   CLASSPATH=d:\peoplesoft\ddp\e920\system\Classes\xalan.jar;d:\peoplesoft\ddp
   \e920\system\Classes\xerces.jar;d:\peoplesoft\ddp\e920\system\Classes
   ```

5. Specify values of the following sections in the jde.ini file:

Section	Property	Description
JDE_CG	`CLASSPATH`	Specify the following path under CLASSPATH: `<JD Edwards EnterpriseOne Installation Directory>\system\Classes`
SECURITY	`SecurityServer`	Specify the name of the security server. The security server is the enterprise server in the JD Edwards EnterpriseOne system with the table F98OWSEC.

Configuring JDBj API Interoperability

To configure JD Edwards EnterpriseOne for JDBj API interoperability, complete the following steps:

1. Generate the serialized spec data source for tables, views, and business functions.
2. Configure `jas.ini`.
3. Configure `JDBj.ini`.

Step 1. Generate the Serialized Spec Data Source

A serialized spec data source consists of serialized spec objects. For JDBj API interoperability, generate the serialized spec objects for tables, views, and the business functions that invoke the batch process.

In JD Edwards EnterpriseOne 9.2 version, serialized spec objects for tables, views, and business functions are generated by default. If serialized spec objects are not generated, you need to generate them.

For more information about generating the serialized spec objects, see JD Edwards EnterpriseOne documentation.

Step 2. Configure the jas.ini File

To configure the JD Edwards EnterpriseOne system environment to support JDBj API interoperability, configure the jas.ini file. For more information, see the "Configuring the jas.ini File" section.

Step 3. Configure the JDBj.ini File

1. Open the JDBj.ini file.
2. Verify that the values of the properties in the JDBj.ini file are correct:

   The JDBj-BOOTSTRAP SESSION section contains the following properties:

   - User
   - Password
   - Environment
   - Role

The JDBj-BOOTSTRAP DATA SOURCE section contains the following properties:

- Name
- DatabaseType
- Server
- ServerPort
- Database
- PhysicalDatabase
- Owner
- Lob
- Unicode
- DatabaseInstance
- TranslateOn

The JDBj-SPEC DATA SOURCE section contains the following properties:

- Name
- DatabaseType
- Server
- ServerPort
- Database
- PhysicalDatabase
- Owner
- Lob
- Unicode
- Password
- DatabaseInstance
- TranslateOn

The JDBj-SERVER section contains the DbcsConversionTolerant property.

The following table lists the JD Edwards EnterpriseOne versions and the entries under the JDBj-SPEC DATA SOURCE section:

JD Edwards EnterpriseOne Version	Entries Under the JDBj-SPEC DATA SOURCE Section	Description
9.2	<Data source information>	Data source refers to the location of serialized spec tables, F989998 and F989999.

If the JDBj-SPEC DATA SOURCE section is not specified, the PowerExchange for JD Edwards EnterpriseOne uses OCM to find the serialized spec tables. If it is not valid, no JD Edwards EnterpriseOne database access is possible.

3. If JD Edwards EnterpriseOne uses an Oracle database, copy the tnsnames.ora file from the machine that hosts JD Edwards EnterpriseOne database to the javalib directory on the PowerCenter Integration Service machine and PowerCenter Client machine.

The following table lists the machines and the javalib directory locations:

Machine	Location
PowerCenter Integration Service	<PowerCenter Installation Directory>\server\bin\javalib

4. Specify the path of the tnsnames.ora file in the Tns property under the JDBj-ORACLE section of the corresponding JDBj.ini file.

   Each machine that hosts the PowerCenter Integration Service and PowerCenter Client has a copy of the JDBj.ini file. For example, the sample code shows the path to the tnsnames.ora file if you copy the tnsnames.ora file in the javalib directory on the PowerCenter Integration Service machine:

   ```
   tns = <PowerCenter Installation Directory>\server\bin\javalib\tnsnames.ora
   ```

   Note: The JDBj.ini file provides instructions for entering values in each section.

Copying JD Edwards EnterpriseOne Libraries and .ini Files

Copy JD Edwards EnterpriseOne API libraries and .ini files to the machines where the PowerCenter Client and the PowerCenter Integration Service are installed. Copy the database driver libraries to the machines where the PowerCenter Integration Service and PowerCenter Client are installed.

Copying API Libraries and .ini Files on the PowerCenter Client Machine

1. Copy the JD Edwards EnterpriseOne API library files to the machine where the PowerCenter Client is installed.

Based on the version of the JD Edwards EnterpriseOne installation directory, copy the files common to the 9.1 and 9.2 installation directories and the files specific to the 9.1 or 9.2 installation directory:

Common files in 9.1 and 9.2 installation directories	File Location
ApplicationAPIs_JAR.jar ApplicationLogic_JAR.jar Base_JAR.jar BIPProxy_JAR.jar BizLogicContainerClient_JAR.jar BizLogicContainer_JAR.jar BusinessLogicServices_JAR.jar castor.jar commons-logging.jar Connector.jar Generator_JAR.jar httpclient.jar httpcore.jar jas.ini jdbj.ini JdbjBase_JAR.jar JdbjInterfaces_JAR.jar JDE.INI jdelog.properties JdeNet_JAR.jar jmxremote_optional.jar ManagementAgent_JAR.jar Metadata.jar MetadataInterface.jar PMApi_JAR.jar Spec_JAR.jar System_JAR.jar SystemInterfaces_JAR.jar xmlparserv2.jar	**Source Location:** <JD Edwards EnterpriseOne <9.1/9.2> Installation Directory>\JDEdwards\<E910/E920> \system\classes **Target Location:** <PowerCenter Installation Directory> \clients\PowerCenter\client\bin\javalib

Note: The files are from JD Edwards EnterpriseOne 9.1 and 9.2 API libraries with 9.1.5 and 9.2 toolsets. For 9.1.5 toolset, the database in JD Edwards EnterpriseOne is Microsoft SQL Server. For 9.2 toolset, the database in JD Edwards EnterpriseOne is Oracle database.

The following table lists the names and locations of the JD Edwards EnterpriseOne 9.1 and 9.2 API library files for the toolset component version 9.1.5 and 9.2 respectively:

Files in 9.1 installation directory	Files in 9.2 installation directory	File Location
commons-codec-1.3.jar commons-httpclient-3.0.jar [1] sqljdbc-1.6.jar xml-apis.jar	commons-codec.jar commons-lang-2.6.jar jmxremote_optional.jar log4j-core.jar ojdbc6.jar tnsnames.ora	**Source Location:**\PeopleSoft\<E910/E920>\system\classes **Target Location:** <PowerCenter Installation Directory>\clients\PowerCenter\client\bin\javalib

1. Copy the commons-httpclient-3.0.jar file from the SchemaGenUtil.zip directory.

2. Copy the jas.ini and jdbj.ini files from the JD Edwards EnterpriseOne thin client installation folder to the following directory on the PowerCenter Client machine:

   ```
   <PowerCenter Installation Directory>\clients\PowerCenter\client\bin\javalib
   ```

   For example, the location of the JD Edwards EnterpriseOne thin client installation folder for Oracle WebLogic is `<WebLogic Installation Directory>\html_server\stage\html_instance\app`.

3. Copy the `jdelog.properties` file from the source to the PowerCenter Client installation directory. The following table lists the name and locations of the file for each version of JD Edwards EnterpriseOne:

JD Edwards EnterpriseOne Version	File Name	File Location
9.1/9.2	jdelog.properties	**Source Location:** <JD Edwards EnterpriseOne <9.1/9.2> Installation Directory>\JDEdwards\<E910/E920>\system\classes\samples\temp\config **Target Location:** <PowerCenter Installation Directory>\clients\PowerCenter\client\bin\javalib

Copying API Libraries and .ini Files on the PowerCenter Integration Service Machine

1. Copy the JD Edwards EnterpriseOne API library files to the nodes on which the PowerCenter Integration Service is installed.

Based on the version of the JD Edwards EnterpriseOne installation directory, copy the files common to the 9.1 and 9.2 installation directories and the files specific to the 9.1 or 9.2 installation directory:

Common files in 9.1 and 9.2 installation directories	File Location
ApplicationAPIs_JAR.jar ApplicationLogic_JAR.jar Base_JAR.jar BIPProxy_JAR.jar BizLogicContainerClient_JAR.jar BizLogicContainer_JAR.jar BusinessLogicServices_JAR.jar castor.jar commons-logging.jar Connector.jar Generator_JAR.jar httpclient.jar httpcore.jar jas.ini jdbj.ini JdbjBase_JAR.jar JdbjInterfaces_JAR.jar JDE.INI jdelog.properties JdeNet_JAR.jar jmxremote_optional.jar ManagementAgent_JAR.jar Metadata.jar MetadataInterface.jar PMApi_JAR.jar Spec_JAR.jar System_JAR.jar SystemInterfaces_JAR.jar xmlparserv2.jar	**Source Location:** <JD Edwards EnterpriseOne <9.1/9.2> Installation Directory>\JDEdwards\<E910/920>\system \classes **Target Location:** <PowerCenter Installation Directory> \server\bin\javalib

The following table lists the names and locations of the JD Edwards EnterpriseOne 9.1 and 9.2 API library files for the toolset component version 9.1.5 and 9.2 respectively:

Files in 9.1 installation directory	Files in 9.2 installation directory	File Location
commons-codec-1.3.jar commons-httpclient-3.0.jar [1] sqljdbc-1.6.jar xml-apis.jar	commons-codec.jar commons-lang-2.6.jar jmxremote_optional.jar log4j-core.jar ojdbc6.jar tnsnames.ora	**Source Location:** <JD Edwards EnterpriseOne <9.1/9.2> Installation Directory>\JDEdwards\<E910/E920>\system\classes **Target Location:** <PowerCenter Installation Directory>\server\bin\javalib

1. Copy the commons-httpclient-3.0.jar file from the SchemaGenUtil.zip directory.

2. If JD Edwards EnterpriseOne uses an Oracle database, copy the tnsnames.ora file from the machine that hosts JD Edwards EnterpriseOne database to the following directory on the PowerCenter Integration Service machine:

   ```
   <PowerCenter Installation Directory>\server\bin\javalib
   ```

3. Copy the jdelog.properties file from the source to the PowerCenter Integration Service installation directory.

   The following table lists the name and locations of the file for each version of JD Edwards EnterpriseOne:

JD Edwards EnterpriseOne Version	File Name	File Location
9.1/9.2	jdelog.properties	**Source Location:** <JD Edwards EnterpriseOne <9.1/9.2> Installation Directory>\JDEdwards\<E910/E920>\system\classes\samples\temp\config **Target Location:** <PowerCenter Installation Directory>\server\bin\javalib

4. Specify the paths to jderoot.log, jas.log, and jasdebug.log log files in the jdelog.properties file. When a JD Edwards EnterpriseOne session uses the APIs, the PowerCenter Integration Service logs the execution summary in the log files.
5. Copy the jas.ini file and jdbj.ini files from the JD Edwards EnterpriseOne thin client installation folder to the following directory on the PowerCenter Integration Service machine:

   ```
   <PowerCenter Installation Directory>\server\bin\javalib
   ```

   For example, the location of the JD Edwards EnterpriseOne thin client installation folder for Oracle WebLogic is `<WebLogic Installation Directory>\html_server\stage\html_instance\app`.

Copying Database Driver Libraries

1. Identify the appropriate database driver libraries. For example:

Underlying Database for JD Edwards EnterpriseOne	Required Libraries
AS400	jt400.jar
Oracle	ojdbc6.jar
UDB	db2java.zip
SQLServer2000	msbase.jar, msutil.jar, and mssqlserver.jar
MS SQLServer2005	sqljdbc.jar
MS SQLServer2008	sqljdbc4.jar

2. On the PowerCenter Integration Service machine, copy the libraries to the following directory:

   ```
   <PowerCenter Installation Directory>\server\bin\javalib
   ```

3. On the PowerCenter Client machine, copy the libraries to the following directory:

   ```
   <PowerCenter Installation Directory>\clients\PowerCenter\client\bin\javalib
   ```

Configuring Access Rights for PowerExchange for JD Edwards EnterpriseOne

You can configure the following types of access rights for PowerExchange for JD Edwards EnterpriseOne:

- Database-level connection. PowerExchange for JD Edwards EnterpriseOne uses a database-level connection to write bulk data into interface tables. The database-level user must exist in the JD Edwards EnterpriseOne database system and this user must have update permissions on interface tables where you want to write data.
- Application-level connection. PowerExchange for JD Edwards EnterpriseOne uses the application-level connection information to connect to the JD Edwards EnterpriseOne system. The application-level user must exist in the JD Edwards EnterpriseOne system. The application-level user must have access to the required JD Edwards EnterpriseOne environment.

  After you connect as an application-level user, PowerExchange for JD Edwards EnterpriseOne uses the database-level API calls to retrieve metadata and the data from the JD Edwards EnterpriseOne database server. Therefore, PowerExchange for JD Edwards EnterpriseOne needs access to the JD Edwards EnterpriseOne system, the object library, the data dictionary, and the business data databases.

  While creating an application-level user, you need to specify a database-level system user. Verify that you provide a database-level system user that has the following database-level access rights:

  - To enable the PowerCenter Integration Service to fetch metadata from the JD Edwards EnterpriseOne system, provide an application-level user name associated with a system user. The system user must have access rights on JD Edwards EnterpriseOne object library tables, control tables, and data dictionary tables.

- To enable the PowerCenter Integration Service to read or write data to the JD Edwards EnterpriseOne system, provide an application-level user name associated with a system user. The system user must have read, insert, update, and delete rights on the tables in the JD Edwards EnterpriseOne environment accessible to the user.

Configuring an ODBC Data Source for JDEOne Load Transformation

Before you run a JDEOne Load transformation, configure an ODBC data source. The Informatica installation includes DataDirect ODBC drivers. If you have existing ODBC data sources created with an earlier version of the drivers, you must create new ODBC data sources using the new drivers. Configure ODBC data sources using the DataDirect ODBC drivers provided by Informatica. For more information on configuring an ODBC data source, see the *Informatica Administrator Guide*.

Troubleshooting Configuration

Registering the plug-in fails.

The PowerExchange for JD Edwards EnterpriseOne plug-in registration fails when the PowerCenter Repository Service already contains a plug-in with the same plug-in ID and vendor ID.

To resolve this issue, complete the following tasks:

1. Unregister the plug-in that has the same plug-in ID and vendor ID as PowerExchange for JD Edwards EnterpriseOne plug-in.
2. Register the PowerExchange for JD Edwards EnterpriseOne plug-in.

The Designer fails to open the Import from JD Edwards EnterpriseOne wizard.

The Designer cannot create Java Virtual Machine (JVM) on the PowerCenter Client machine if the PowerCenter Client machine does not have enough allocated memory. When this occurs, the JDEOWBulkClientLogC.log file displays the following error:

```
Unable to create JVM
```

The JDEOWBulkClientLogC.log file is located at the following path:

```
<PowerCenter Installation Directory>\clients\PowerCenter\client\bin
```

Verify that the PowerCenter Client machine has at least 64 MB of allocated memory for JVM. If you increase the allocated memory and get the same error, set the allocated memory to a higher value.

Complete the following steps to configure the allocated memory on the PowerCenter Client machine:

1. Create the JVMOptions.ini file in the following directory:

   ```
   <PowerCenter Installation Directory>\clients\PowerCenter\client\bin
   ```
2. Enter the Xmx property in the JVMOptions.ini file and set it to 64, 128, 256, or 512 MB.
   For example, you can enter:

   ```
   Xmx=64
   ```

Note: The Xmx property sets the maximum allocated memory. If you get the same error after configuring the property, set it to a higher value.

CHAPTER 3

JD Edwards EnterpriseOne Sources

This chapter includes the following topics:

Creating a DSN

A DSN contains connection information to connect to the JD Edwards EnterpriseOne environment. The information includes elements such as JD Edwards EnterpriseOne user name and password.

You can create a DSN for each environment you want to connect to. After you create a DSN, you cannot view the DSN attributes or edit the DSN.

1. In the Source Analyzer, click Sources > Import JD Edwards OneWorld Source.

   The Import JD Edwards OneWorld Source dialog box appears.
2. Click the Browse button.

   The DSN Creation dialog box appears.
3. Enter the following information:

DSN Attribute	Description
DSN	Name for the DSN.
User Name	JD Edwards EnterpriseOne user name for establishing the connection.
Password	Password for the user name.
Environment	Name of the JD Edwards EnterpriseOne environment you want to connect to.
Role	Role of the JD Edwards EnterpriseOne user.

4. Click Save.

   The Designer connects to the JD Edwards EnterpriseOne environment. After successful connection, the Designer saves the DSN.

Importing a JD Edwards EnterpriseOne Source Definition

When you connect to JD Edwards EnterpriseOne to import a JD Edwards EnterpriseOne source definition, the Designer displays tables and views that you can import. The Designer also displays the custom tables in JD Edwards EnterpriseOne. The tables and views are grouped by system codes.

Working with Filters

You can enter a filter to reduce the number of tables or views that appear in the Import JD Edwards OneWorld Source dialog box. If you do not enter a filter, all the tables and views appear in the dialog box.

You can enter a filter condition by using an SQL expression or a Perl compatible regular expression syntax.

The following table describes the characters that can be used in a regular expression to specify a valid filter condition:

Character	Description
X	Any character in a Perl compatible regular expression.
\	Backslash character. It introduces escaped constructs and quotes characters that otherwise would be interpreted as un-escaped constructs. For example, the expression \ \ matches a single backslash and \{ matches a left brace.
.	Single character. Similar to _ in SQL.
.*	Zero or more characters. Similar to % in SQL.

The following table describes the character classes that can be used in a regular expression to specify a valid filter condition:

Character Classes	Description
[abc]	a, b, or c (simple class).
[^abc]	Any character except a, b, or c (negation).

The following table describes the qualifiers that can be used in a regular expression to specify a valid filter condition:

Greedy Qualifiers	Description
X?	X once or not at all.
X*	X zero or more times.
X+	X one or more times.
X{n}	X exactly n times.
X{n,}	X at least n times.
X{n,m}	X at least n but not more than m times.

The following table describes the logical operators that can be used in a regular expression to specify a valid filter condition:

Logical Operators	Description
XY	X followed by Y.
X|Y	Either X or Y.
(X)	X as a group of characters to be matched.

Invalid Characters

The following characters are not valid in an SQL expression:

```
! # $ * / ( ) + , ^ < = > ? @ ; & . : { | }
```

The following characters are not valid in a regular expression:

```
! # / , < = > @ ; : % _
```

Importing a Source Definition

1. In the Source Analyzer, click Sources > Import JD Edwards OneWorld Source.

   The Import JD Edwards OneWorld Source dialog box appears.
2. In DSN, select a DSN Name from the available list.
3. Enter the password.
4. Optionally, enter a filter.
5. From the Apply Filter list, select Business Table or Business View.
6. To filter tables, from the On list, select Table Name or Table Description. Or, to filter views, select View Name or View Description.
7. Click Connect or Reconnect to apply the filter.

   A list of tables appears. To see the list of views, click the Views tab.

8. To add tables or views to import as a source definition, select the tables or views under an application, and then click Add to Import List.

   To select all tables or views relevant to a particular application, select the application and then click Select All. To clear the selected tables or views under a particular application, click Select None.

9. Optionally, to view either the list of selected tables and views or both, click View Import List.

   The JD Edwards OneWorld Source Import List dialog box appears.

10. To remove tables and views from the source import list, select the tables and views, and click Remove.

    The modified source import list appears.

11. Click OK to import the tables and views or either of them as source definitions.

    The Designer adds underscore (_) as a prefix to the column names that start with a number or an at symbol (@) in the JD Edwards EnterpriseOne system.

Editing JD Edwards EnterpriseOne Source Definition

After you import JD Edwards EnterpriseOne source definitions, you can edit the source definition to modify the datatype and the key type fields. Edit the JD Edwards EnterpriseOne source definition in the following situations:

- To change the Math_Numeric datatype to Jdetime
- To extract data from more than one table or view

Working with the Jdetime Datatype

When you import a source definition, the Designer imports columns of the Jdetime datatype as Math_Numeric. To process the data correctly during a session, you must edit the source definition and manually change the datatype to Jdetime. Edit the datatype of a column on the Columns tab of a source definition.

Creating a User-Defined Join

To read data from two or more tables, join the JD Edwards EnterpriseOne source definitions by defining the primary key-foreign key relationship between them. To define the relationship, define a primary key in one table and the corresponding foreign key in another table.

To define the key type, change the Key Type attribute to PRIMARY KEY and FOREIGN KEY on the Columns tab.

When you define primary and foreign keys, the Source Analyzer connects the foreign key field to the primary key field with an arrow.

Troubleshooting JD Edwards EnterpriseOne Sources

PowerExchange for JD Edwards EnterpriseOne is not enabled on PowerCenter or the license is expired. Contact Informatica Global Customer Support.

The license key does not contain an entry for the JD Edwards EnterpriseOne plug-in. Or, the license for the JD Edwards EnterpriseOne plug-in is expired.

Verify that the repository contains a valid license for PowerExchange for JD Edwards EnterpriseOne.

PowerExchange for JD Edwards EnterpriseOne plug-in is not available in the repository.

JD Edwards EnterpriseOne plug-in is not registered in the repository. Verify that the JD Edwards EnterpriseOne plug-in is registered in the repository.

It takes a long time to connect to JD Edwards EnterpriseOne when I click Connect from the Import JD Edwards OneWorld Source dialog box.

Cancel the operation. Remove the unwanted files from the output directory at JD Edwards EnterpriseOne Enterprise Server. The unwanted files are specified as the XML List output directory in jde.ini file. Try to connect to JD Edwards EnterpriseOne again.

While applying a filter condition, an error message indicating an invalid filter appears.

A filter condition can be either in the SQL format or in the pure regular expression format. You cannot specify both formats in a single filter condition. You cannot have a single filter condition that contains the following formats:

- '.' and '_'
- '.' and '%'
- '.*' and '%'

CHAPTER 4

Application Source Qualifier

This chapter includes the following topics:

Application Source Qualifier Overview

Configure the following SQL properties for the Application Source Qualifier:

- Join Type. Specify how to join sources.
- Number of Sorted Ports. Specify the number of sorting ports.
- Select Distinct. Extract distinct rows.
- Source Filter. Filter data that you want to extract from JD Edwards EnterpriseOne sources.

Configure SQL properties on the SQL Editor tab.

Join Type

Specify the join type to join JD Edwards EnterpriseOne sources. By default, tables are joined by the inner join. You can specify the following types of joins to override the default join type:

- Left outer. Performs a left outer join on JD Edwards EnterpriseOne tables on which the join is defined.
- Right outer. Performs a right outer join on JD Edwards EnterpriseOne tables on which the join is defined.

Note: The options in the Join Type field are not case sensitive.

Number of Sorted Ports

Sort the extracted rows from a JD Edwards EnterpriseOne source by specifying the number of ports. If multiple sources are attached to a single Application Source Qualifier, you can sort the rows extracted from the sources. If you enter zero or do not enter any value, the PowerCenter Integration Service does not sort data.

Distinct Data

You can configure the Application Source Qualifier to extract distinct rows from a single JD Edwards EnterpriseOne source. Specify one of the following values:

- Yes. Extracts distinct rows.
- No. Extracts duplicate rows.

Note: The values are not case sensitive.

Source Filter

Specify a filter condition to reduce the number of rows you want to extract from a JD Edwards EnterpriseOne source. You can specify more than one filter condition to extract data. You can also include parameters and variables in a filter condition. The PowerCenter Integration Service handles parameters and variables in the following order:

1. Predefined server variables
2. Mapping parameters and variables
3. Workflow variables
4. Session parameters

Note: When you declare the variables and parameters for the Jdetime and Jdedate datatypes, specify the datatype as “string.”

Syntax for the Filter Condition

Use the following syntax when you enter a source filter.

- To compare a field in a JD Edwards EnterpriseOne table with any other field, use the following syntax:

  ```
  <Tablename.columnname> <operator> <Tablename.columnname>
  ```

- To compare a field in a JD Edwards EnterpriseOne table with a literal value, use the following syntax:

  ```
  <Tablename.columnname> <operator> <literal>
  ```

- To compare a field in a JD Edwards EnterpriseOne table with a range of values, use one of the following syntaxes:

  ```
  <Tablename.columnname> BETWEEN <literal1> AND <literal2>
  <Tablename.columnname> NOT BETWEEN <literal1> AND <literal2>
  ```

  Note: The <Tablename.columnname> BETWEEN <literal1> and <literal2> format is equivalent to [<Tablename.columnname> >= <literal1>] AND [<Tablename.columnname> <= <literal2>]

- To compare a field in a JD Edwards EnterpriseOne table with a set of values, use one of the following syntaxes:

  ```
  <Tablename.columnname> IN (literal1,literal2,literal3, …..)
  <Tablename.columnname> NOT IN (literal1,literal2,literal3, …..)
  ```

You must specify the database table name as Tablename while specifying the filter condition to extract data from a table.

Note: You can use an underscore (_) or a period (.) to separate table name and column name.

Operators

The following table describes the operators that you can use in a filter condition:

Operator	Description
<	Extracts data where value of a field is lesser than the value of a literal or the other field. For example, F0101.AN8 < 100.
>	Extracts data where value of a field is greater than the value of a literal or the other field. For example, F0102.AN9 > F0104.AX5.
=	Extracts data where value of a field is equal to a literal or the other field. For example, F0102.AN9 = F0104.AX5. You can also compare strings using this operator.
<=	Extracts data where value of a field is lesser than or equal to the value of a literal or the other field. For example, F0102.AN9 <= 405.
>=	Extracts data where value of a field is greater than or equal to the value of a literal or the other field. For example, F0102.AN9 >= 208.
!=	Extracts data where value of a field is not equal to the value of a literal or the other field. For example, F0102.AN9 != 2435. You can also compare strings using this operator.
AND, and	Extracts data that satisfies more than one filter condition. For example, use the following filter condition to extract data for the employees who stay in U.S. and whose salary is less than $200: [F005.Location = 'U.S.'] AND [F005.SAL < 200]
OR, or	Extracts data that satisfies any one of the specified filter conditions. For example, use the following filter condition to extract data for the employees who either stay in U.S. or U.K.: [F005.Location = 'U.S.'] OR [F005.Location = 'U.K.']
LIKE, like	Extracts the string values that match a particular pattern. For example, use the following filter condition to extract data of the employees whose names start with Ace: F1010.Empname LIKE 'Ace%' The LIKE operator is not case sensitive.
BETWEEN..AND, between..and	Extracts data from a range of values. For example, use the following filter condition to extract data for those employees whose salary is between $200 and $500: F0001.SAL BETWEEN 200 AND 500 The filter condition is equivalent to [F0001.SAL >= 200] AND [F0001.SAL <= 500]
NOT BETWEEN ..AND, not between.. and	Extracts data where the value of a field is not from a range of values. For example, use the following filter condition to extract data for those employees whose salary is not between $100 and $200: F0001.SAL NOT BETWEEN 100 AND 200

Operator	Description
IN, in	Extracts data where value of a field is a member of a set of valid values. For example, use the following filter condition to extract all the rows where value of AN12 is 101, 102, or 103: F0001.AN12 IN (101, 102, 103)
NOT..IN, not..in	Extracts data where value of a field is not a member of a set of valid values. For example, use the following filter condition to extract all the rows where value of BN19 is not 101 or 102: F0001.AN12 NOT IN (101, 102)

Note: Query validation and result set depends on the database on which JD Edwards EnterpriseOne is installed.

Rules and Guidelines for Filter Conditions

Use the following guidelines when you enter a filter condition:

- The table name in a filter condition must be the name of the table imported from JD Edwards EnterpriseOne. If the filter is specified on a view, get the table name from the column name in the view source definition.
- When you specify a source filter, delete the underscore (_) from the field name that starts with an at sign (@) or a number.
- The String values should be enclosed in single quotes.
- Enter the literal value based on the datatype of the column specified in the first token. For example,

```
F0101.AN8 < 100
F0101.ALPH LIKE 'MARKETTING COMPANY'
```

- Specify the literal value in the YYYY-MM-DD format if the datatype is Jdedate. For example,

```
F0101.UPMJ = 2003-05-29
```

- Specify the literal value in the HH:MM:SS format if the datatype is Jdetime. For example,

```
F0101_UPMT = 15:10:25
```

- Use the AND and OR operators to enter more than one filter condition.
- Separate filter conditions by square brackets ([ ]).
- Comparison operators have higher precedence than the logical operators AND and OR.
- AND operator has higher precedence over the OR operator.
- Use square brackets ([ ]) to change the precedence of operators.

Note: You cannot apply source filter on a binary field.

CHAPTER 5

JD Edwards EnterpriseOne Targets

This chapter includes the following topics:

JD Edwards EnterpriseOne Target Overview

JD Edwards EnterpriseOne target definitions represent metadata for JD Edwards EnterpriseOne tables. To import a JD Edwards EnterpriseOne target definition, create a DSN to establish a connection with JD Edwards EnterpriseOne Enterprise Server.

When you import a target definition, you can filter the tables you want the Designer to display. After you import a target definition, you can edit the target definition.

Importing a JD Edwards EnterpriseOne Target Definition

When you connect to JD Edwards EnterpriseOne to import a JD Edwards EnterpriseOne target definition, the Designer displays tables that you can import. The Designer also displays the custom tables in JD Edwards EnterpriseOne. The tables are grouped by system codes.

You can enter a filter to reduce the number of tables that the Import JD Edwards OneWorld Target dialog box displays. If you do not enter a filter, all the available tables in the JD Edwards EnterpriseOne system appear in the Designer.

Importing a Target Definition

1. In the Target Designer, click Targets > Import JD Edwards OneWorld Target.

   The Import JD Edwards OneWorld Target dialog box appears.
2. In DSN, select a DSN Name from the available list.
3. Enter a password.
4. Optionally, enter a filter.
5. From the Apply Filter list, select Business Table.
6. From the On list, select Table Name or Table Description.
7. Click Connect or Reconnect to apply the filter.

   A list of tables appears.
8. To import tables as a target definition, select the tables under an application, and then click Add to Import List.
9. To view the list of tables selected for importing, click View Import List.

   The JD Edwards OneWorld Target Import List dialog box appears.
10. Optionally, select single or multiple tables to remove from the target import list, and click Remove.

    The modified target import list appears.
11. To import the tables as target definitions, click OK.

Editing JD Edwards EnterpriseOne Target Definition

PowerCenter can identify columns with Jdedate datatype, but cannot identify columns with Jdetime datatype. When you import a target definition that uses a Jdetime datatype, the Designer imports the target definition with the Math_Numeric datatype. You need to identify these columns and change the datatype from Math_Numeric to Jdetime.

Troubleshooting JD Edwards EnterpriseOne Targets

PowerExchange for JD Edwards EnterpriseOne is not enabled on PowerCenter or the license is expired. Contact Informatica Global Customer Support.

The license key does not contain an entry for the JD Edwards EnterpriseOne plug-in. Or, the license for the JD Edwards EnterpriseOne plug-in is expired.

Verify that the repository contains a valid license for PowerExchange for JD Edwards EnterpriseOne.

PowerExchange for JD Edwards EnterpriseOne plug-in is not available in the repository.

JD Edwards EnterpriseOne plug-in is not registered in the repository. Verify that the JD Edwards EnterpriseOne plug-in is registered in the repository.

It takes a long time to connect to JD Edwards EnterpriseOne when I click Connect from the Import JD Edwards OneWorld Target dialog box.

Cancel the operation. Remove the unwanted files from the output directory at JD Edwards EnterpriseOne Enterprise Server. The unwanted files are specified as the XML List output directory in jde.ini file. Try to connect to JD Edwards EnterpriseOne again.

CHAPTER 6

JDEOne Load Transformation

This chapter includes the following topics:

JDEOne Load Transformation Overview

The JDEOne Load transformation is an active transformation that you can use to transform data from corporate data sources, and then load the data into staging tables called interface tables. JD Edwards EnterpriseOne interface tables mirror JD Edwards EnterpriseOne application tables. JD Edwards EnterpriseOne provides predefined interface tables for some applications. You can also create custom interface tables that are formatted based on JD Edwards EnterpriseOne standards.

When you run a session with a JDEOne Load transformation, the PowerCenter Integration Service connects to a JD Edwards EnterpriseOne database server and loads data into the interface tables. If you run the session and the JDEOne Load transformation encounters error rows, the JDEOne Load transformation passes the error rows as output.

JDEOne Load Transformation Components

A JDEOne Load transformation contains the following tabs:

- Transformation. Enter the name and description of the transformation. The naming convention for a JDEOne Load transformation is JLT_*TransformationName*. You can also make the transformation reusable.
- Ports. View ports on the Ports tab.
- Properties. View the properties on the Properties tab. Configure the tracing level property for the JDEOne Load transformation.
- Edit Interface Tables. View and edit the interface table metadata on the Edit Interface Tables tab. If a transformation is a reusable transformation, you cannot edit the transformation in the Mapping Designer.

Configuring JDEOne Load Transformation Properties

You can configure the tracing level of the session log on the Properties tab.

JDEOne Load Transformation Groups and Ports

A JDEOne Load transformation can contain multiple input groups. Each input group maps to an interface table and contains ports that correspond to columns in the interface table.

A JDEOne Load transformation also contains the following output groups:

- Status_Output
- Control_Fields
- Error_Output

The field name is stored in the Business Name attribute of the port. The description of the port contains more information about the port.

Status_Output Group

The Status_Output group contains a row for the load status of each partition, regardless of the number of input groups.

If the load for the entire partition was successful, the Status port displays 1. The PowerCenter Integration Service counts the number of non-fatal errors that occur while loading data into the interface table. If the error count for a partition reaches the error count limit, the PowerCenter Integration Service fails the load and sets the status to 0 for that partition. If the error count for a partition reaches the error threshold, the PowerCenter Integration Service fails the session.

Control_Fields Group

The Control_Fields group contains a row for each unique combination of control fields and interface table. Control fields contain information that uniquely identify each row that was successfully inserted or updated in the interface table. The PowerCenter Integration Service uses control fields when inserting, updating, or deleting data in the interface table. The naming convention for Control_Fields ports is <group name>_<field name>.

This group contains the following ports:

- Control_Fields_ZTABLE. Contains the name of the interface table into which the PowerCenter Integration Service loads data.
- Control_Fields_ ZTABLE_ALIAS. Contains the alias of the interface table into which the PowerCenter Integration Service loads data.
- Control_Fields_EDUS. Contains the user ID that the business function uses to run the batch process or the User ID of the person who receives messages in the Employee Work Center.
- Control_Fields_EDBT. Contains the batch number for the rows that the PowerCenter Integration Service loads into the interface table.

To improve session performance, PowerExchange for JD Edwards EnterpriseOne does not maintain or pass control field information if the Control_Fields group is not linked in the mapping.

Error_Output Group

If an error occurs while loading data into a row in the interface table, the PowerCenter Integration Service passes the erroneous row to the Error group. The naming convention for Error_Output ports is <error>_<field name>.

This group contains the following ports:

- Error_ZTABLE. Contains the name of the interface table into which the PowerCenter Integration Service is loading data.
- Error_EDUS. Contains the user ID that the business function uses to run the batch process or the User ID of the person who receives messages in the Employee Work Center.
- Error_EDBT. Contains the batch number for the rows that the PowerCenter Integration Service failed to load into the interface table.
- Error_EDTN. Contains the transaction number.
- Error_EDLN. Contains the ID that is used to identify a record within a transaction.
- Error_MSG_Error. Contains detailed error messages.

Using a JDEOne Load Transformation in a Mapping

When using a JDEOne Load transformation in a mapping, link the ports in the source with the corresponding ports in the JDEOne Load transformation.

Rules and Guidelines for JDEOne Load Transformations

Use the following rules and guidelines when you create a JDEOne Load transformation in a mapping:

- Ensure that you link all ports that are imported as a primary key. If you do not link all ports that are imported as a primary key, the PowerCenter Integration Service does not insert, update, or delete rows in the JD Edwards EnterpriseOne database.
- If you link one of the ports that belongs to the Control_Fields group, ensure that you link all other ports that belong to the Control_Fields group. If you link one, but not all ports that belong to the Control_Fields group, the Designer marks the mapping as not valid.

Creating a JDEOne Load Transformation

You can create a JDEOne Load transformation in the Transformation Developer or the Mapping Designer.

1. Open the Transformation Developer. Or, open the Mapping Designer.
2. Click Transformation > Create.

   The Create Transformation dialog box appears.
3. Select JDEOne Load from the list.
4. Enter a name for the JDEOne Load transformation.
5. Click Create.

   The wizard appears.

6. Enter information for the following connection parameters:

Connection Parameter	Description
User Name	User name to connect to JD Edwards EnterpriseOne .
Password	Password to connect to JD Edwards EnterpriseOne.
Environment	JD Edwards EnterpriseOne environment to which you want to connect.
Role	Role of the JD Edwards EnterpriseOne user.

7. Click Connect.
8. Click Next.
9. To view the list of interface tables in JD Edwards EnterpriseOne , click Get Objects. Or, to view a specific list of interface tables, enter a filter condition in the Filter Value field, select Table Name or Table Description from the Filter On list, and then click Get Objects.

   Note: By default, you cannot view interface tables that do not follow the standard naming convention. To search for all types of interface tables, enter a filter condition, and then select the Search Through All Tables check box.
10. Select the interface tables that you want to import.

    If you select interface tables each time you apply a filter, the Designer imports all the interface tables that you selected for every filter. If you do not want to import an interface table, enter the filter condition for that interface table, and then clear the check box for that interface table.
11. Click Next.

    If you selected interface tables from different applications, a confirmation message appears. To import all the selected interface tables in this JDEOne Load transformation, click Yes.
12. Select the master interface table from the Select Master Interface Table list.

    Note: The Designer imports the master interface table as the first input group.
13. Optionally, you can click an interface table in the Selected Interface Table area to view the interface table metadata.
14. Optionally, you can update the following interface table metadata in the Interface Table Fields area:
    - Primary key details
    - Datatype for columns that have the Math_Numeric, Jdetime, or Jdedate datatype. You can change Math_Numeric to Jdetime or Jdedate, Jdetime to Math_Numeric, or Jdedate to Math_Numeric.
15. Click Next.
16. Click Finish.

Understanding the Load Summary for the JDEOne Load Transformation

When the PowerCenter Integration Service loads data into the JDE EnterpriseOne Interface tables, it writes the load summary in the session log for each group in a partition. The load summary lists the following types of row counts for each group in a partition:

- Requested rows. Number of rows that the PowerCenter Integration Service receives from the input group to load data into the JDE EnterpriseOne Interface tables.
- Applied rows. Number of rows that the PowerCenter Integration Service tries to write to the JDE EnterpriseOne Interface tables.
- Rejected rows. Number of rows that the PowerCenter Integration Service does not load into the JDE EnterpriseOne Interface tables because of an error.
- Inserted rows. Number of rows that the PowerCenter Integration Service inserts into the JDE EnterpriseOne Interface tables.
- Updated rows. Number of rows that the PowerCenter Integration Service updates in the JDE EnterpriseOne Interface tables.
- Deleted rows. Number of rows that the PowerCenter Integration Service deletes from the JDE EnterpriseOne Interface tables.
- Unprocessed rows. Number of rows that the PowerCenter Integration Service does not process in the group because it reaches the error count limit for the partition.

CHAPTER 7

JDEOne Batch Process Invoker Transformation

This chapter includes the following topics:

JDEOne Batch Process Invoker Transformation Overview

The JDEOne Batch Process Invoker transformation is an active transformation that you can use to execute a business function that invokes the batch process. The batch process triggers the data flow from an interface table to a JD Edwards EnterpriseOne base table.

By default, the JDEOne Batch Process Invoker transformation executes the LaunchBatchApplication business function. You can configure the JDEOne Batch Process Invoker transformation to invoke a different business function.

JDEOne Batch Process Invoker Transformation Components

A JDEOne Batch Process Invoker transformation contains the following tabs:

- Transformation. Enter the name and description of the transformation. The naming convention for a JDEOne Batch Process Invoker transformation is JIT_TransformationName. You can also make the transformation reusable.
- Ports. View ports on the ports tab.
- Properties. View the properties on the Properties tab. Configure the tracing level property for the JDEOne Batch Process Invoker transformation.

- Edit Batch Process Invoker BSFN. View and edit the datatype, precision, and scale of the JDEOne Batch Process Invoker transformation parameters. If a transformation is a reusable transformation, you cannot edit the transformation in the Mapping Designer.

JDEOne Batch Process Invoker Transformation Groups and Ports

A JDEOne Batch Process Invoker transformation contains the following input and output groups:

- Status input group
- Input_Param_Group input group
- Output_Param_Group output group

Status Group

The Status group contains the Status port, which indicates whether the PowerCenter Integration Service previously was able to successfully load data into the interface tables. If the status is set to 1, the PowerCenter Integration Service successfully loaded data into interface tables. When the status is set to 1, the JDEOne Batch Process Invoker transformation executes the business function that invokes the batch process. If the status is set to 0, the PowerCenter Integration Service was unable to load the data into the interface tables, and, therefore, the JDEOne Batch Process Invoker does not execute the business function.

To make the JDEOne Batch Process Invoker transformation execute the business function based on the status of the interface table load, link the output status port of the JDEOne Load transformation to the input status port of the JDEOne Batch Process Invoker transformation.

Input_Param_Group Group

The Input_Param group contains input parameters that are required to execute the specified business function. The input parameters may vary depending on the business function or the JD Edwards EnterpriseOne version. For information about the input parameters for a business function, see the JD Edwards EnterpriseOne documentation.

Note: You can invoke only one batch for every invoker transformation. The PowerCenter Integration Service considers only the first row received for the Input_Param_Group in a batch process invocation and ignores subsequent rows.

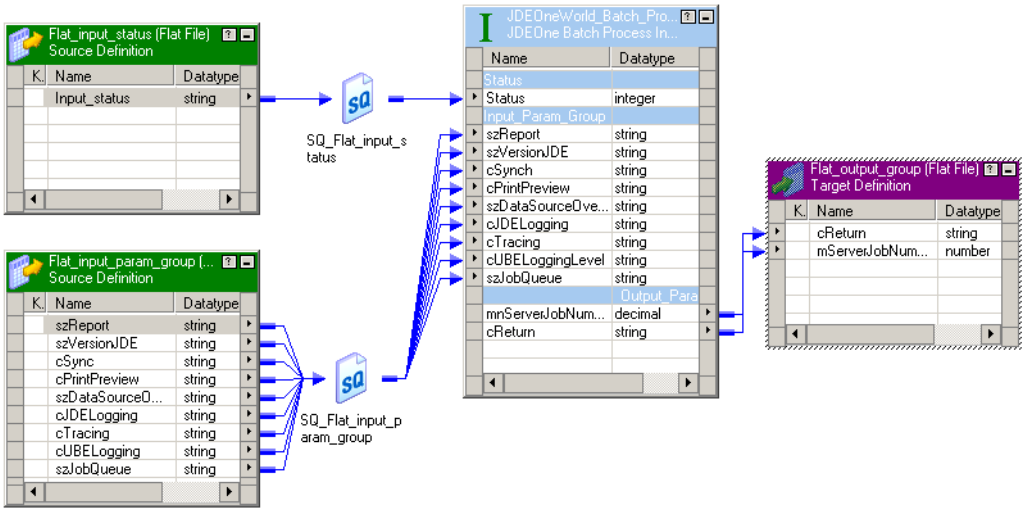
Output_Param_Group Group

The Output_Param_Group group returns output parameters after the business function submits the batch process to the Job Queue system. You can use output parameters for other tasks. For example, you can provide the status of the batch process as an output parameter and perform another task based on the status. The output parameters vary depending on the business function or the JD Edwards EnterpriseOne version. For information about the output parameters for a business function, see the JD Edwards EnterpriseOne documentation.

JDEOne Batch Process Invoker Transformation in a Mapping

You can use the JDEOne Batch Process Invoker transformation independently or with the JDEOne Load transformation. When using a JDEOne Batch Process Invoker transformation in a mapping, link the ports in the source that contains the parameter values with the corresponding ports in the JDEOne Batch Process Invoker transformation.

The following figure shows a mapping with a JDEOne Batch Process Invoker transformation that runs the specified business function that invokes the batch process:



The batch process loads the data from the interface table to the base table.

The mapping contains the following components:

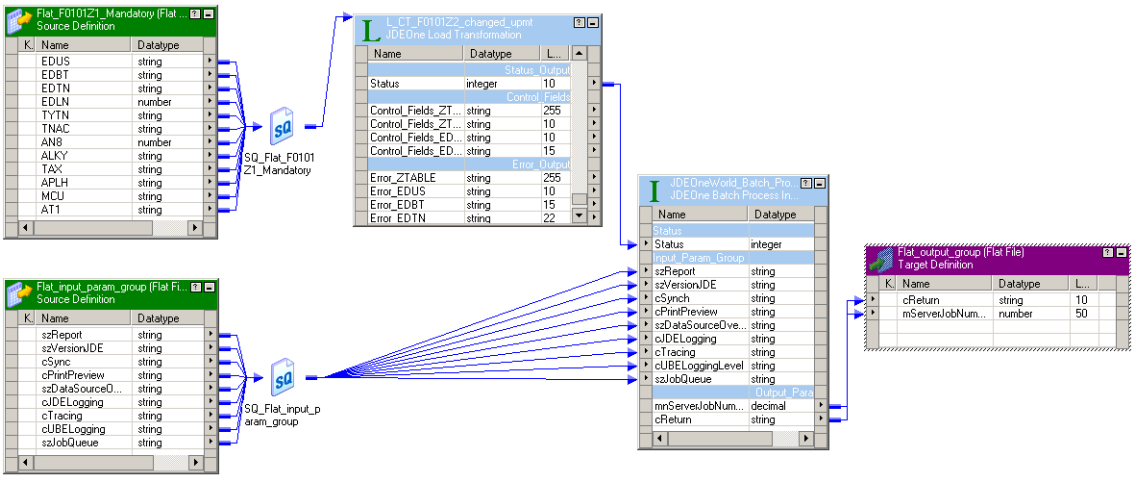
- Flat_input_status. A source flat file that provides a value to the Status port of the JDEOne Batch Process Invoker transformation.
- SQ_Flat_input_status. An Application Source Qualifier that links the input status in the source flat file to the Status port of the JDEOne Batch Process Invoker transformation.
- Flat_input_param_group. A source flat file that provides the input parameter values that are required to execute the specified business function.
- SQ_Flat_input_param_group. An Application Source Qualifier that links the input parameters in the source flat file to the corresponding input parameters in the JDEOne Batch Process Invoker transformation.
- JDEOneWorld_Batch_Process. A JDEOne Batch Process Invoker transformation used in the mapping to execute the specified business function that invokes the batch process.
- Flat_output_group. After the specified business function invokes the batch process, the PowerCenter Integration Service writes the output status to the flat file target.

JDEOne Batch Process Invoker Transformation with the JDEOne Load Transformation

When using a JDEOne Batch Process Invoker transformation with a JDEOne Load transformation in a mapping, map the parameter values in the source ports to the corresponding ports in the JDEOne Batch Process Invoker transformation. To execute the business function based on the status of the interface table

load, link the Status output port of the JDEOne Load transformation with the Status input port of the JDEOne Batch Process Invoker transformation.

The following figure shows a mapping where a JDEOne Batch Process Invoker transformation is used with a JDEOne Load transformation:



The mapping contains the following components:

- Flat_F0101Z1_Mandatory. A source flat file that contains the data to be loaded into the specified JD Edwards EnterpriseOne interface table.
- SQ_Flat_F0101Z1_Mandatory. An Application Source Qualifier that links the ports in the source flat file to the ports in the input group of the JDEOne Load transformation.
- L_CT_F0101Z2_changed_upmt. A JDEOne Load transformation used in the mapping to load data from the source into the specified JD Edwards EnterpriseOne interface table. The output status port of this JDEOne Load transformation is linked to the input status port of the JDEOne Batch Process Invoker transformation.
- Flat_input_param_group. A source flat file that provides the input parameter values that are required to execute the specified business function.
- SQ_Flat_input_param_group. An Application Source Qualifier that links the input parameters in the source flat file to the corresponding input parameters in the JDEOne Batch Process Invoker transformation.
- JDEOneWorld_Batch_Process. A JDEOne Batch Process Invoker transformation used in the mapping to execute the specified business function that invokes the batch process.
- Flat_output_group. After the specified business function invokes the batch process, the PowerCenter PowerCenter Integration Service writes the output status to the flat file target.

Creating a JDEOne Batch Process Invoker Transformation

You can create a JDEOne Batch Process Invoker transformation in the Transformation Developer or the Mapping Designer. If you want the JDEOne Batch Process Invoker transformation to execute a different

business function, you can update the business function name while creating the JDEOne Batch Process Invoker transformation.

1. Open the Transformation Developer. Or, open the Mapping Designer.
2. Click Transformation > Create.

   The Create Transformation dialog box appears.
3. Select JDEOne Batch Process Invoker from the list.
4. Enter a name for the JDEOne Batch Process Invoker transformation.
5. Click Create.

   The wizard appears.
6. Enter the information for the following connection parameters:

Connection Parameter	Description
User Name	User name to connect to JD Edwards EnterpriseOne.
Password	Password to connect to JD Edwards EnterpriseOne.
Environment	JD Edwards EnterpriseOne environment to which you want to connect.
Role	Role of the JD Edwards EnterpriseOne user.

7. Click Connect.
8. Click Next.
9. To execute a different business function to invoke the batch process, type the name of the business function in the BSFN Name box.

   By default, the JDEOne Batch Process Invoker transformation executes the LaunchBatchApplication business function to invoke the batch process.
10. Click Show MetaData.

    The parameter details appear. You can edit the following parameter details.

Parameter Detail	Description
Parameter Name	A parameter name can contain a maximum of 80 characters. You can use an '_' (underscore) in a parameter name. You cannot have duplicate parameter names.
Type	Set the parameter type to Input or Output.
Precision	You can edit the data type for parameters that have the char, string, varstring, and longvarbinary data types.
Scale	You can edit the scale for parameters that have math_numeric data type.
Data Type	You can edit the data type for each parameter.

11. Click Next.
12. Click Finish.

CHAPTER 8

JD Edwards EnterpriseOne Sessions

This chapter includes the following topics:

Configuring Application Connections

Before you run a JD Edwards EnterpriseOne session, configure a JD Edwards EnterpriseOne connection for JD Edwards EnterpriseOne sources, targets, or transformations in the Workflow Manager.

PowerExchange for JD Edwards EnterpriseOne Connections

When you configure a JD Edwards EnterpriseOne connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the JD Edwards EnterpriseOne.

The following table describes the connection properties:

Property	Description
User Name	User name to connect to JD Edwards EnterpriseOne database.
Password	Password for the user name.
Connect String	Name of the ODBC data source. **Note:** Use Informatica certified ODBC drivers for ODBC data source connection.

Property	Description
Connection Retry Period	Number of seconds that the PowerCenter Integration Service waits after making a request to connect to the database. If the PowerCenter Integration Service does not receive any response, the session fails. Default value is 0.
Control Table Name Prefix	Owner of the F0005 control table that contains UDC values. If the database user specified in the database connection is not the owner of the F0005 control table and the session is configured for UDC validation, specify the owner of the F0005 control table as the control table name prefix. You can use a parameter for this connection attribute.

Configuring an Application Connection for a JDEOne Load Transformation

JDEOne Load transformations use a JDEOne Loader Connection to connect to the JD Edwards EnterpriseOne database server.

Note: For more information about connection parameters, contact your JD Edwards EnterpriseOne system administrator.

1. In the **Workflow Manager**, click **Connections** > **Application**.

   The **Application Connection Browser** dialog box appears.
2. Click **New**.
3. Select **JDEOne Loader Connection** from the list.
4. Click **OK**.

   The **Connection Object Definition** dialog box appears.
5. Enter the JD Edwards EnterpriseOne connection properties.
6. Click **OK**.

Configuring an Application Connection

The JD Edwards EnterpriseOne sources, targets, and JDEOne Batch Process Invoker transformations use a JD Edwards EnterpriseOne connection to connect to the JD Edwards EnterpriseOne repository.

1. In the Workflow Manager, click Connections > Application.

   The Application Connection Browser dialog box appears.
2. Click New.
3. Select JDEOneWorld Connection from the list.
4. Click OK.

   The Connection Object Definition dialog box appears.

5. Enter the following information in the connection object:

Connection Attribute	Description
User Name	User name to connect to JD Edwards EnterpriseOne application.
Password	Password for the user name.
Environment	JD Edwards EnterpriseOne environment you want to connect to.
Role	Role of the JD Edwards EnterpriseOne application user.

6. Click OK.

   Note: You can view load statistics in the session log. The load summary in the Workflow Monitor does not display load statistics.

Configuring Sessions for a JDEOne Load Transformation

You can configure the following session properties for a JDEOne Load transformation.

The following table describes the session properties for a JDEOne Load transformation:

Session Property	Description
Enable Data Conversion	Select this option if you want the data to be converted before the data is inserted into the interface table.
Error Count Limit	Enter a limit for the error count. Set to '0' to disable the limit. Valid values include 0 to 2^31-1. Default is 0.
Truncate Table	Select to truncate the interface table before the PowerCenter Integration Service loads data into this interface table.
Enable UDC Validation	Select to validate the data against the list of user-defined code (UDC) values that correspond to the UDC type. By default, the PowerCenter Integration Service does not validate the data in the interface table.
Interface Table Load Type	Select one of the following load types: - Single Row. The PowerCenter Integration Service loads the data row-by-row into interface tables. - Multiple Rows. The PowerCenter Integration Service loads the data for multiple rows in the same sequence as that of the source rows. The PowerCenter Integration Service processes consecutive rows with the Insert row type in batches, and processes rows with the Update or Delete row types one row at a time. Default is Multiple Rows.

Session Property	Description
Skip Row on Conversion Error	Select this option to skip records with erroneous data. When a record is skipped, the Error count and the Error threshold is incremented by 1. If you do not select this option, the PowerCenter Integration Service inserts null values for fields that have data conversion errors. In this case, the PowerCenter Integration Service increments the error count by 1.
Table Name Prefix	Specify the owner of the interface table if the database user specified in the database connection is not the owner of the Interface table. Specify the table name prefix in the following format: `<Table name>:<Prefix>` To specify a prefix for multiple interface tables, separate each prefix by a comma.

Configuring the Load Type for the JDEOne Load Transformation

You can configure the load type for the JDEOne Load transformation. The load type determines if the PowerCenter Integration Service loads single rows or multiple rows.

When you configure the load type for the JDEOne Load transformation as single rows, the PowerCenter Integration Service loads the data row-by-row into interface tables.

When you configure the load type for the JDEOne Load transformation as multiple rows, the PowerCenter Integration Service loads the data into interface tables in batches based on the buffer block size. If you configure the session to treat source rows as data-driven and the load type as multiple rows, the PowerCenter Integration Service loads data into interface tables based on the following conditions:

- If all source rows have the row type Insert, the PowerCenter Integration Service creates batches to load data into interface tables. The PowerCenter Integration Service loads the rows into interface tables in batches based on the buffer block size.
- If all source rows have the row type Update or Delete, the PowerCenter Integration Service loads the data row-by-row into interface tables.
- If a source contains rows with multiple row types, the PowerCenter Integration Service loads the rows into interface tables in the same sequence as that of the source rows. The PowerCenter Integration Service processes rows with the Update or Delete row type on a row-by-row basis. The PowerCenter Integration Service loads consecutive rows with the Insert row type in batches, based on the buffer block size.

For example, you configure the session to treat source rows as data driven and the load type as multiple rows.

The following table displays the rows contained in the source:

Row ID	Row Type
1	Update
2	Insert
3	Insert
4	Update
5	Delete

Row ID	Row Type
6	Insert
7	Update

The PowerCenter Integration Service processes rows in the following sequence:

1. Row with row ID 1.
2. Batch A that contains rows with the row type as Insert.

   The following table displays Batch A that contains rows with the row type as Insert:

Batch ID	Row ID
A	2
A	3

3. Row with row ID 4.
4. Row with row ID 5.
5. Batch B that contains one row with the row type as Insert:

   The following table displays Batch B that contains one row with the row type as Insert:

Batch ID	Row ID
B	6

6. Row with row ID 7.

Validating Data against User-defined Code Values

You can enable UDC validation before the PowerCenter Integration Service writes data to the interface table. The PowerCenter Integration Service can validate the incoming field value against user-defined code values. If the incoming value is not a valid code value, the PowerCenter Integration Service does not write the row in the interface table. The PowerCenter Integration Service marks the row as an erroneous row and increases the error limit and error threshold counts by 1.

For example, the interface table F0101ZX contains a column CTR that contains country codes. The corresponding code values are stored under the control table F0005 for SY='00' and RT='CN' combination.

If you enable UDC validation, the PowerCenter Integration Service verifies the incoming value for the CTR column with the possible country codes stored in the column DRKY in control table F0005 for the SY='00' and RT='CN' combination. If the PowerCenter Integration Service finds a matching value, the PowerCenter Integration Service loads that data into the interface table. Otherwise, the PowerCenter Integration Service writes the erroneous rows into the ERROR output group of the JDEOne Load transformation.

Configuring Sessions for a JDEOne Batch Process Invoker Transformation

You can select the Abort Session on Fatal Target Error session property to abort the session when the PowerCenter Integration Service encounters a fatal error.

Handling Multiple Partitions when Using the JDEOne Batch Process Invoker Transformation

If a session contains multiple partitions, the JDEOne Batch Process Invoker transformation consolidates the data for all partitions into the first partition, also called the master partition. The JDEOne Batch Process Invoker transformation invokes the batch process only after it receives data for all the groups and partitions.

Note: You cannot run the session on a grid if the corresponding mapping contains a JDEOne Batch Process Invoker transformation

The PowerCenter Integration Service uses the parameters in the first row of the Input_Param_Group. It ignores parameters received for other partitions. The PowerCenter Integration Service generates the output for Output_Param_Group for the consolidated data in the master partition.

Error Handling

When you run a session, the PowerCenter Integration Service can encounter fatal or non-fatal errors.

Error Handling for Fatal Errors

When you enable the Abort Session on Fatal Target Error session property, the PowerCenter Integration Service terminates the session when any of the following fatal errors occur:

- Connection errors
- Failure to invoke the batch process

If a session timeout occurs, the JD Edwards EnterpriseOne server continues to run the batch process even if the PowerCenter Integration Service terminates the session. However, the PowerCenter Integration Service does not write the batch process ID to the session log.

Error Handling for Non-Fatal Errors

If a non-fatal error occurs, the PowerCenter Integration Service increments the error count for each non-fatal error. The PowerCenter Integration Service continues to process the session until the PowerCenter Integration Service reaches the error count limit or error count threshold, or the session completes. If the error count reaches the error count limit, the PowerCenter Integration Service does not process subsequent rows and sets the output status of the JDEOne Load transformation to 0. If the error count reaches the error threshold, the PowerCenter Integration Service fails the session. When the PowerCenter Integration Service reaches the error count limit or error count threshold, it does not roll back the previously loaded rows.

When the PowerCenter Integration Service loads rows into an interface table in batches, it compares the error count against the error count limit and threshold after it processes each batch.

Troubleshooting the JD Edwards EnterpriseOne Sessions

The session fails when Oracle is the underlying source database, and you select the Enable UDC Validation session property.

When PowerExchange for JD Edwards EnterpriseOne uses the DataDirect 6.0 drivers for an Oracle source connection, and you select the Enable UDC Validation session property, the PowerCenter Integration Service reads the JD Edwards EnterpriseOne control table F0005. If the JD Edwards EnterpriseOne control table F0005 contains data in an Oracle column with the NCHAR datatype, the session fails.

To resolve this issue, select the Enable N-Char Support session property in the data source name (DSN) definition that uses the DataDirect 6.0 driver for Oracle.

The session can truncate data when Oracle is the underlying database, and you select the Enable N-Char Support and the Enable UDC Validation options.

The session truncates data written to the interface table when all the following statements are true:

- The PowerExchange for JD Edwards EnterpriseOne session uses a DataDirect 6.0 driver to connect to an Oracle target.
- You select the Enable N-Char Support and Enable UDC Validation options in the DSN definition for the Oracle target.
- The Oracle table contains columns with the NCHAR datatype.

To resolve this issue, complete one of the following tasks:

- For UNIX, add the ColumnSizeAsCharacter attribute, and set the value of the DSN property in the ODBC.ini file to 1.
- For Windows, add the ColumnSizeAsCharacter attribute, and set the value of the registry entry in the DSN definition to 1.

APPENDIX A

Data Type Reference

This appendix includes the following topic:

JD Edwards EnterpriseOne and Transformation Data Types

PowerCenter uses the following data types in JD Edwards EnterpriseOne mappings:

- JD Edwards EnterpriseOne native data types. JD Edwards EnterpriseOne data types appear in the JD Edwards EnterpriseOne definitions in a mapping.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the PowerCenter Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration Service reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the PowerCenter Integration Service writes to a target, it converts the transformation data types to the comparable native data types.

The following table describes the JD Edwards EnterpriseOne data types that PowerCenter supports, and the corresponding transformation data types:

JD Edwards EnterpriseOne Data Type	Description	Transformation Data Type	Description
Char	Precision 1	String	1 to 104,857,600 characters
Id_Long	Precision 11, integer value	Bigint	9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0, integer value
Int	Precision 4	Integer	Precision 10, scale 0
Jdedata	Jan 1, 1900 A.D. to Dec 31, 2899 A.D., precision 6	date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to nanosecond)

JD Edwards EnterpriseOne Data Type	Description	Transformation Data Type	Description
Jdetime	Timestamp in Coordinated Universal Time (UTC), precision 6	date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to nanosecond)
Jdeutime	Precision 11	date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to nanosecond)
LongVarBinary	Precision depends on the source field and the underlying database for JD Edwards EnterpriseOne	Binary	1 to 104,857,600 bytes
Math_Numeric	Precision 1 to 28 digits, scale 0 to 28	Decimal	Precision 1 to 28 digits, scale 0 to 28
String	1 to 104,857,600 characters	String	1 to 104,857,600 characters
Varstring	1 to 104,857,600 characters	String	1 to 104,857,600 characters

Jdetime Data Type

PowerCenter can identify columns with Jdedate data type, but cannot identify columns with Jdetime data type. When you import a source or a target definition that uses a Jdetime data type, the Designer imports the source or the target definition with the Math_Numeric data type. You need to identify these columns and change the data type from Math_Numeric to Jdetime.

Note: Not every column of the Math_Numeric data type is imported from the Jdetime data type. Before you change the data type of a Math_Numeric column, verify that the source or the target in the JD Edwards EnterpriseOne uses a Jdetime data type.

APPENDIX B

Error Messages

This appendix includes the following topics:

Designer Messages

The Designer displays messages in message boxes and in the output window.

JdeSocket error: Connection timed out.

Explanation: The JD Edwards EnterpriseOne Enterprise Server service is not started or the machine is not running.

User Response: Verify that the machine that hosts the JD Edwards EnterpriseOne Enterprise Server is running and the JD Edwards EnterpriseOne Enterprise Server service is started.

PowerCenter Integration Service Messages

[BSFN_CALLOBJECT_ERROR]

An error occurred during the call to trigger business function <Business_Function_Name>: COSE#1010 Remote environment initialization failed.

User Response: The user login has expired or is disabled for the JD Edward EnterpriseOne environment you want to access.

Explanation: Specify a valid user login. Contact your system administrator to enable the user login or create a user login.

[DATA_SOURCE_NOT_FOUND]

Data source for F00950, TBLE not found. ([OCM_DEFAULT_ENTRY_NOT_FOUND] The OCM does not contain a default entry for environment <environment name>).

Explanation: An invalid environment is specified in the application connection.

User Response: Specify a valid environment.

[ERROR] **The function <Function_Name> in module <Module_Name> is configured to run LOCAL. LOCAL is not supported.**

Explanation: The session run is using JD Edwards EnterpriseOne sources or targets with dependent business functions configured to execute on the local machine. The local machine does not have the libraries required to execute the business functions.

User Response: Configure the JD Edwards EnterpriseOne system to execute the business functions on the JD Edwards EnterpriseOne logic server that has the required libraries. Use the JD Edwards EnterpriseOne Object Configuration Manager for the required configurations.

Verify that the JD Edwards EnterpriseOne Object Configuration Manager contains an entry for the function. If no entry exists, you can either specify 'Default' or an entry for the function for the object type 'BSFN.' Verify that this entry is associated with the primary data source for the JD Edwards EnterpriseOne environment that you are accessing. The primary data source is a Server data source hosted on the JD Edwards EnterpriseOne logic server.

Error: [SPEC_NOT_FOUND]

Unable to find EnterpriseOne specification for TABLE-<table name>.

Explanation: A serialized specification data object is not generated for the selected table or view.

User Response: Use the JD Edwards EnterpriseOne eGenerator application to generate the serialized specification object.

[SECURITY_ERROR2]

A security error occurred, status=1, error code=337.

Explanation: An invalid user name is specified in the application connection.

User Response: Specify a valid user name.

[SECURITY_ERROR2]

A security error occurred, status=2, error code=331.

Explanation: An invalid password is specified in the application connection.

User Response: Specify a valid password.

[44065] **The PowerCenter Integration Service failed to allocate memory to the ERROR_OUTPUT group buffer.**

Explanation: The PowerCenter Integration Service encountered a memory allocation error.

User Response: Close some applications that are running on the PowerCenter Integration Service machine and try again.

[44066] **The PowerCenter Integration Service failed to allocate memory to the CONTROL_FIELDS group buffer.**

Explanation: The PowerCenter Integration Service encountered a memory allocation error.

User Response: Close some applications that are running on the PowerCenter Integration Service machine and try again.

[44067] **The PowerCenter Integration Service failed to allocate memory to the STATUS_OUTPUT group buffer.**

Explanation: The PowerCenter Integration Service encountered a memory allocation error.

User Response: Close some applications that are running on the PowerCenter Integration Service machine and try again.

[44068] The Enable Data Conversion check box is not selected. The PowerCenter Integration Service may fail to prepare or bind a statement for the field <field name>.

Explanation: The JDEDATE and JDETIME datatypes require data conversion. The PowerCenter Integration Service may fail to prepare or bind a statement for the field if the Enable Data Conversion check box is not selected.

User Response: Select the Enable Data Conversion check box for the JDETIME or JDEDATE datatype, or specify the converted data in the field.

[44075] The PowerCenter Integration Service is inserting a null value for field <field name> because the Skip Row on Conversion Error check box is not selected.

Explanation: The Skip Row on Conversion Error check box is not selected and the year for the JDEDATE datatype exceeds the maximum value 2899.

User Response: Verify that the year for a JDEDATE column is less than or equal to 2899.

[44076] The PowerCenter Integration Service failed to load the row because the Skip Row on Conversion Error check box is selected.

Explanation: The Skip Row on Conversion Error check box is selected and the year for the JDEDATE datatype exceeds the maximum value 2899.

User Response: Verify that the year for the JDEDATE column is less than or equal to 2899.

[4013_InputStatusFailure_Exception]

The PowerCenter Integration Service will not execute business function <business function name> to invoke the batch process because the PowerCenter Integration Service previously failed to load the corresponding interface tables.

Explanation: The PowerCenter Integration Service will not invoke the business function until it successfully loads the corresponding interface tables.

User Response: Resolve the issues with the Load transformation that loads the corresponding interface tables.

[4021_JDBjInvBSFNErr]

The PowerCenter Integration Service failed to execute the business function <business function name> through the JDBj APIs.

Explanation: The required API libraries are not copied to the following directory: <PowerCenter Installation Directory>/server/bin/javalib, the serialized specification object is not generated for the business function, or the JD Edwards EnterpriseOne server encountered an internal error while executing the business function.

User Response: Verify that the required API libraries are copied to the following directory <PowerCenter Installation Directory>/server/bin/javalib and the serialized specification object is generated for the business function.

[4037_WrongDataType]

The imported STATUS port does not contain the Integer datatype.

Explanation: The datatype changed after importing the STATUS port or an error occurred while converting the datatype to Integer.

User Response: Verify that the datatype did not change.

[4042 _NoInParamData]

The PowerCenter Integration Service will not execute the business function <business function name> to invoke the batch process because it did not receive a value for the INPUT_PARAM_GROUP group for partition <partition number>.

Explanation: The INPUT_PARAM_GROUP group did not receive values for the connected input parameters.

User Response: Verify that the source row that is linked to the INPUT_PARAM_GROUP group contains data.

APPENDIX C

Glossary

JD Edwards EnterpriseOne Enterprise Server

The server component in JD Edwards EnterpriseOne. JD Edwards EnterpriseOne can have a primary and a secondary enterprise server. However, the PowerCenter Integration Service only communicates with the primary enterprise server.

JD Edwards EnterpriseOne environment

A business environment to access application suites in JD Edwards EnterpriseOne. You can connect to different business environments such as Production, Pristine, and Development. JD Edwards EnterpriseOne identifies each business environment with an environment name. For example, for JD Edwards EnterpriseOne 8.10, the production environment is identified as environment PD810.

system codes

A unique code for an application suite in JD Edwards EnterpriseOne. For example, the application suites Accounts Receivable and Accounts Payable each have a system code. When you create a business table or view for an application suite, the system code is attached to the table or view.

INDEX

M

P

R

S

T

V