



Informatica® PowerExchange for JD  
Edwards EnterpriseOne  
10.2.2

# User Guide

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2019-02-21

# Table of Contents

<b>Preface .....</b>	<b>5</b>
Informatica Resources. ....	5
Informatica Network. ....	5
Informatica Knowledge Base. ....	5
Informatica Documentation. ....	5
Informatica Product Availability Matrices. ....	6
Informatica Velocity. ....	6
Informatica Marketplace. ....	6
Informatica Global Customer Support. ....	6
 <b>Chapter 1: Understanding PowerExchange for JD Edwards EnterpriseOne.....</b>	 <b>7</b>
Understanding PowerExchange for JD Edwards EnterpriseOne Overview. ....	7
Understanding JD Edwards EnterpriseOne. ....	7
JD Edwards EnterpriseOne Tables and Views. ....	8
JD Edwards EnterpriseOne Business Functions. ....	8
Handling Bulk Data Movement in JD Edwards EnterpriseOne. ....	8
Informatica Integration with JD Edwards EnterpriseOne. ....	9
 <b>Chapter 2: PowerExchange for JD Edwards EnterpriseOne Configuration.....</b>	 <b>11</b>
PowerExchange for JD Edwards EnterpriseOne Configuration Overview. ....	11
Prerequisites. ....	11
Copying JD Edwards EnterpriseOne Libraries and .ini Files. ....	11
Copy the API Libraries and .ini Files. ....	12
Copy the Database Driver Libraries. ....	13
 <b>Chapter 3: JD Edwards EnterpriseOne Connections.....</b>	 <b>15</b>
JD Edwards EnterpriseOne Connection Overview. ....	15
JD Edwards EnterpriseOne Connection Properties. ....	16
Creating a JD Edwards EnterpriseOne Connection in the Developer Tool. ....	17
Creating an JD Edwards EnterpriseOne Connection in the Administrator Tool. ....	18
 <b>Chapter 4: JD Edwards EnterpriseOne Data Objects.....</b>	 <b>19</b>
JD Edwards EnterpriseOne Data Objects Overview. ....	19
JD Edwards EnterpriseOne Data Object Properties. ....	20
JD Edwards EnterpriseOne Data Object Read Operation Properties. ....	21
Output Properties of a Data Object Read Operation. ....	21
Rules and Guidelines for Filter Conditions. ....	24
JD Edwards EnterpriseOne Data Object Write Operation Properties. ....	25
Input Properties of a JD Edwards EnterpriseOne Data Object Write Operation. ....	25
Target Properties of a JD Edwards EnterpriseOne Data Object Write Operation. ....	25

JD Edwards EnterpriseOne Data Object Interface Table Write Operation Properties. . . . .	26
Input Properties of a JD Edwards EnterpriseOne Data Object Interface Table Write Operation. .	27
Importing a JD Edwards EnterpriseOne Data Object. . . . .	28
Configuring Filters to Import a JD Edwards EnterpriseOne Data Object. . . . .	28
Editing JD Edwards EnterpriseOne Source Objects. . . . .	30
Creating a JD Edwards EnterpriseOne Data Object Operation. . . . .	30
Rules and Guidelines for JD Edwards EnterpriseOne Objects. . . . .	30
Troubleshooting a Filter. . . . .	31
<b>Chapter 5: JD Edwards EnterpriseOne Mappings. . . . .</b>	<b>32</b>
JD Edwards EnterpriseOne Mappings Overview. . . . .	32
JD Edwards EnterpriseOne Mapping Read Example. . . . .	33
JD Edwards EnterpriseOne Mapping Interface Table Write Example. . . . .	34
<b>Appendix A: Data Type Reference. . . . .</b>	<b>36</b>
JD Edwards EnterpriseOne and Transformation Data Types. . . . .	36
Jdetime Data Type. . . . .	37
<b>Index. . . . .</b>	<b>39</b>

# Preface

The *Informatica PowerExchange® for JD Edwards EnterpriseOne User Guide* provides information about reading data from or writing data to a JD Edwards EnterpriseOne server. It is written for database administrators and developers who create mappings to read from or write data from a JD Edwards EnterpriseOne resource.

This book assumes that you have knowledge of JD Edwards EnterpriseOne and Informatica.

## 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](mailto: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>.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at <https://search.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at [infa\\_documentation@informatica.com](mailto: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](mailto: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, 7](#)
- [Understanding JD Edwards EnterpriseOne, 7](#)
- [Handling Bulk Data Movement in JD Edwards EnterpriseOne, 8](#)
- [Informatica Integration with JD Edwards EnterpriseOne, 9](#)

## 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 Data Integration Service uses JDBC 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

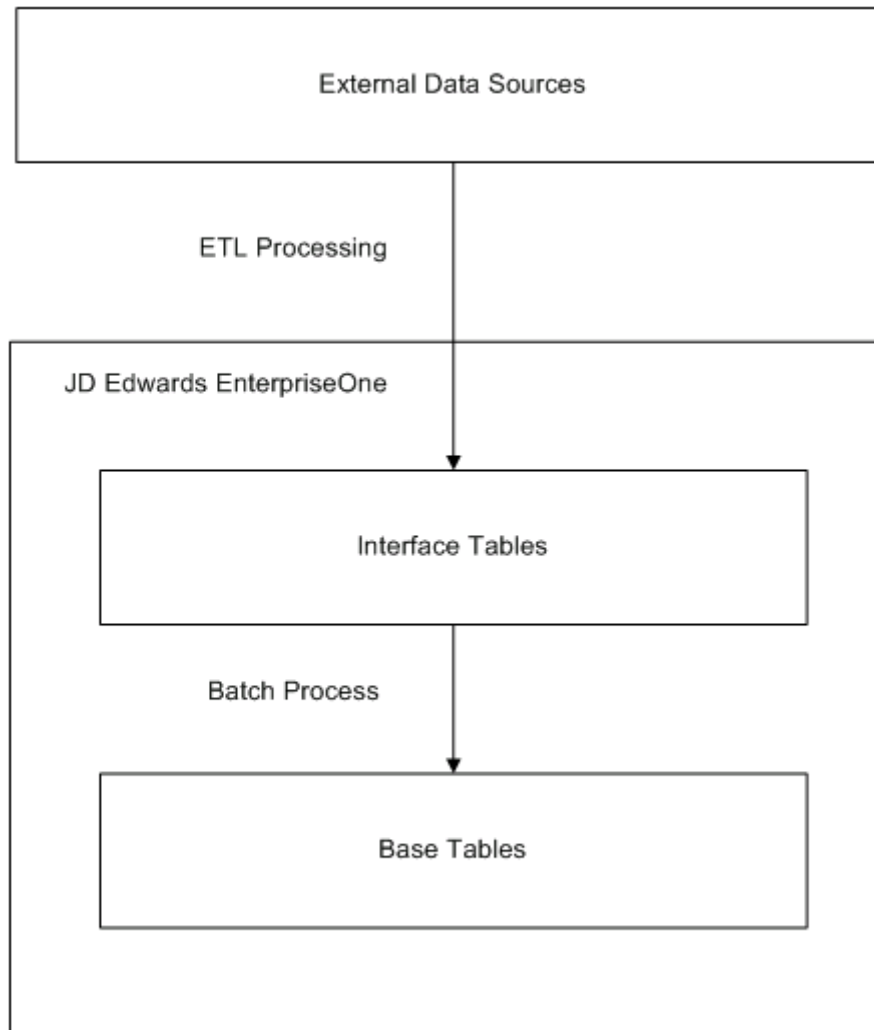
You can use PowerExchange for JD Edwards EnterpriseOne to load data in bulk to JD Edwards EnterpriseOne targets.

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:

- The Data Integration Service loads data from external data sources into working tables called interface tables. JD Edwards EnterpriseOne interface tables mirror JD Edwards EnterpriseOne application tables.
- The Data Integration Service runs the business function that invokes the batch process. The batch process triggered from the function extracts data from the interface tables, validates the data, and writes that data to the base tables.



The following figure shows the bulk load architecture:



## Informatica Integration with JD Edwards EnterpriseOne

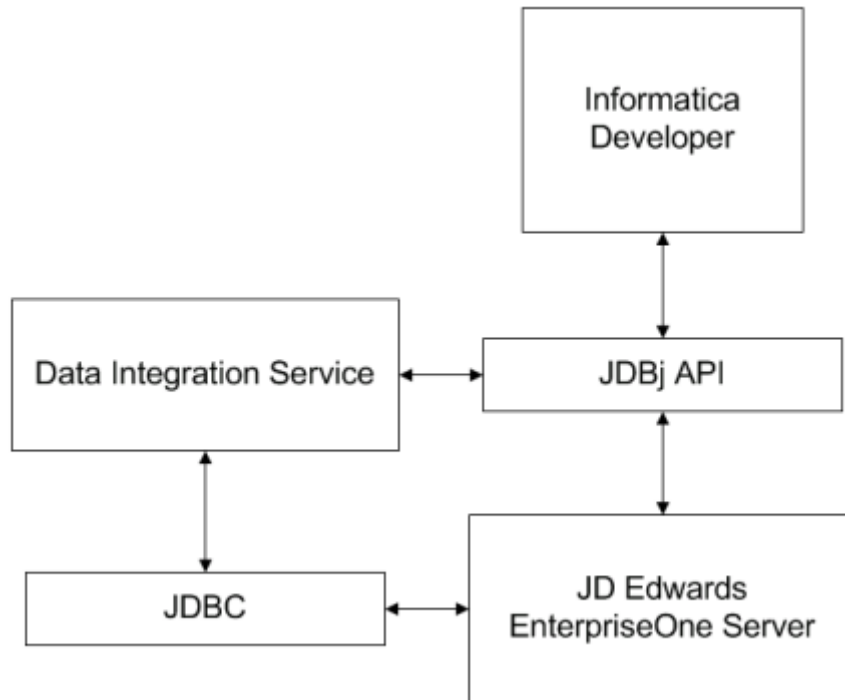
Before you connect to the JD Edwards EnterpriseOne Enterprise Server, you must configure the JD Edwards EnterpriseOne Enterprise Server to support interoperability using XML and the JDBj APIs.

You can import JD Edwards EnterpriseOne tables and views as source or target data objects. When you import a source or target data object, Informatica connects to a JD Edwards EnterpriseOne Enterprise Server to import the metadata.

When Informatica connects to JD Edwards EnterpriseOne, it connects to an environment. The Developer Tool 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 data object.

When you run a mapping, the Data Integration Service connects to JD Edwards EnterpriseOne to read data from sources and write data to targets. When the Data 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 Data Integration Service integrates with JD Edwards EnterpriseOne:



The Data Integration Service reads and writes data based on the mapping and application connection configuration in Informatica Developer. 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 Data Integration Service uses the JDBj APIs to read data from or write data to JD Edwards EnterpriseOne after establishing a connection with the specified environment. The Data Integration Service uses the JDBC connection string to write data in bulk to the JD Edwards EnterpriseOne interface table.

## CHAPTER 2

# PowerExchange for JD Edwards EnterpriseOne Configuration

This chapter includes the following topics:

- [PowerExchange for JD Edwards EnterpriseOne Configuration Overview, 11](#)
- [Copying JD Edwards EnterpriseOne Libraries and .ini Files, 11](#)

## PowerExchange for JD Edwards EnterpriseOne Configuration Overview

PowerExchange for JD Edwards EnterpriseOne is installed with the Informatica Services.

### Prerequisites

Before you use PowerExchange for JD Edwards EnterpriseOne, install and configure Informatica services and clients.

Create the following services in the Informatica domain:

- Data Integration Service
- Model Repository Service

## Copying JD Edwards EnterpriseOne Libraries and .ini Files

Copy the JD Edwards EnterpriseOne API libraries, .ini files, and database driver libraries to the machines where you installed the Informatica Client and the Data Integration Service.

## Copy the API Libraries and .ini Files

Ensure that all the libraries are available in the Informatica Client and Server machines for proper functioning of PowerExchange for JD Edwards EnterpriseOne.

- Copy the JD Edwards EnterpriseOne API library files to the machines where you installed the Informatica Client and Data Integration Service.

The following table lists the names and locations of the JD Edwards EnterpriseOne 9.1.5 API library files with 9.1.5 toolset where the database in JD Edwards EnterpriseOne is Microsoft SQL Server:

File Name	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-codec-1.3.jar commons-httpclient-3.0.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 ManagementAgent_JAR.jar Metadata.jar MetadataInterface.jar PMApi_JAR.jar Spec_JAR.jar sqljdbc-1.6.jar System_JAR.jar SystemInterfaces_JAR.jar xerces.jar xml-apis.jar xmlparserv2.jar	<b>Source Location:</b> <JD Edwards EnterpriseOne 9.1 Installation Directory>\JDEdwards\E910\system\classes <b>Target Informatica Client Location:</b> <Informatica Installation Directory>\clients\DeveloperClient\connectors\thirdparty\infa.jdeel\common <b>Target Data Integration Service Location:</b> <Informatica Installation Directory>\connectors\thirdparty\infa.jdeel\common

The following table lists the names and locations of the JD Edwards EnterpriseOne 9.1.4 API library files with 9.1.4 toolset where the database in JD Edwards EnterpriseOne is Oracle:

File Name	File Location
ApplicationAPIs_JAR.jar ApplicationLogic_JAR.jar Base_JAR.jar BizLogicContainerClient_JAR.jar BizLogicContainer_JAR.jar BusinessLogicServices_JAR.jar castor.jar commons-codec-1.3.jar commons-logging.jar Connector.jar Generator_JAR.jar httpclient.jar httpcore.jar jas.ini jdbj.ini JdbjBase_JAR.jar JdbjInterfaces_JAR.jar JdeNet_JAR.jar jmxremote_optional.jar log4j.jar ManagementAgent_JAR.jar Metadata.jar MetadataInterface.jar PMApi_JAR.jar Spec_JAR.jar System_JAR.jar SystemInterfaces_JAR.jar tnsnames.ora xmlparserv2.jar	<p><b>Source Location:</b> &lt;JD Edwards EnterpriseOne 9.1 Installation Directory&gt;\JDEdwards\E910\system\classes</p> <p><b>Target Informatica Client Location:</b> &lt;Informatica Installation Directory&gt;\clients\DeveloperClient\connectors\thirdparty\infa.jdeel\common</p> <p><b>Target Data Integration Service Location:</b>&lt;Informatica Installation Directory&gt;\connectors\thirdparty\infa.jdeel\common</p>

## Copy the Database Driver Libraries

Copy the database driver libraries to the machines where you have installed the Informatica Server and Client.

1. Identify the appropriate database driver libraries. For more information about the JDBC jar libraries specific to the database that you use, see the JD Edwards EnterpriseOne installation documentation.

For example, the following table lists the libraries required for JD Edwards EnterpriseOne database:

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 Data Integration Service machine, copy the libraries to the following directory:  
`<Informatica Installation Directory>\connectors\thirdparty\infa.jdeel`
3. On the Informatica Client machine, copy the libraries to the following directory:  
`<Informatica Installation Directory>\clients\DeveloperClient\connectors\thirdparty\infa.jdeel`

## CHAPTER 3

# JD Edwards EnterpriseOne Connections

This chapter includes the following topics:

- [JD Edwards EnterpriseOne Connection Overview, 15](#)
- [JD Edwards EnterpriseOne Connection Properties, 16](#)
- [Creating a JD Edwards EnterpriseOne Connection in the Developer Tool, 17](#)
- [Creating an JD Edwards EnterpriseOne Connection in the Administrator Tool, 18](#)

## JD Edwards EnterpriseOne Connection Overview

Create a connection to import JD Edwards EnterpriseOne metadata to create data objects, preview data, and run mappings.

Configure a JD Edwards EnterpriseOne connection before the Data Integration Service can read data from the JD Edwards EnterpriseOne sources or write data to the JD Edwards EnterpriseOne targets.

You can create a JD Edwards EnterpriseOne connection in the Developer tool, the Administrator tool, or through the `infacmd isp` command.

# JD Edwards EnterpriseOne Connection Properties

Use a JD Edwards EnterpriseOne connection to connect to a JD Edwards EnterpriseOne object.

The following table describes the JD Edwards EnterpriseOne connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * ( ) - + = { [ ] }   \ : ; " ' < , > . ? /
ID	The string that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 765 characters.
Location	The Informatica domain where you want to create the connection.
Type	The connection type. Select JD Edwards EnterpriseOne.
Host Name	JD Edwards EnterpriseOne server host name.
Enterprise Port	JD Edwards EnterpriseOne server port number. Default is 6016.
User Name	The JD Edwards EnterpriseOne database user name.
Password	The password for the JD Edwards EnterpriseOne database user.
Environment	Name of the JD Edwards EnterpriseOne environment you want to connect to.
Role	Role of the JD Edwards EnterpriseOne user. Default is *ALL.
User Name	The JD Edwards EnterpriseOne database user name.
Password	Password for the database user.



Property	Description
Driver Class Name	<p>The following list provides the driver class name that you can enter for the applicable database type:</p> <ul style="list-style-type: none"> <li>- DataDirect JDBC driver class name for Oracle: <code>com.informatica.jdbc.oracle.OracleDriver</code></li> <li>- DataDirect JDBC driver class name for IBM DB2: <code>com.informatica.jdbc.db2.DB2Driver</code></li> <li>- DataDirect JDBC driver class name for Microsoft SQL Server: <code>com.informatica.jdbc.sqlserver.SQLServerDriver</code></li> </ul> <p>For more information about which driver class to use with specific databases, see the vendor documentation.</p>
Connection String	<p>The connection string to connect to the database. Use the following connection string:</p> <p>The JDBC connection string uses the following syntax:</p> <ul style="list-style-type: none"> <li>- For Oracle: <code>jdbc:informatica:oracle://&lt;host name&gt;:&lt;port&gt;,ServiceName=&lt;db service name&gt;</code></li> <li>- For DB2: <code>jdbc:informatica:db2://&lt;host name&gt;:&lt;port&gt;;databaseName=&lt;db name&gt;</code></li> <li>- For Microsoft SQL: <code>jdbc:informatica:sqlserver://&lt;host name&gt;:&lt;port&gt;;databaseName=&lt;db name&gt;</code></li> </ul>

## Creating a JD Edwards EnterpriseOne Connection in the Developer Tool

Create a connection before you import JD Edwards EnterpriseOne data objects, preview data, or run mappings.

1. Click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain.
4. Select **Enterprise Applications > JD Edwards EnterpriseOne** and click **Add**.
5. Enter a connection name.
6. Enter an ID for the connection.
7. Optionally, enter a connection description.
8. Select the domain where you want to create the connection.
9. Select **JD Edwards EnterpriseOne** as the connection type.
10. Click **Next**.
11. Configure the connection properties.
12. Click **Test Connection** to verify that you can connect to the JD Edwards EnterpriseOne server.  
The Data Integration Service validates the JDBC parameters only at run-time when you write data in bulk to JD Edwards EnterpriseOne server and not when you test the connection.
13. Click **Finish**.

# Creating an JD Edwards EnterpriseOne Connection in the Administrator Tool

Create a connection before you import JD Edwards EnterpriseOne data objects, preview data, or run mappings.

1. In the Administrator tool, click the **Domain** tab.
2. Click the **Connections** view.
3. In the Navigator, select the domain.
4. In the Navigator, click **Actions > New > Connection**.

The New Connection dialog box appears.

5. In the **New Connection** dialog box, select JD Edwards EnterpriseOne, and then click **OK**.

The **New Connection** wizard appears.

6. Enter a connection name.
7. Enter an ID for the connection.
8. Optionally, enter a connection description.
9. Enter the connection properties
10. Click **Test Connection** to verify that you can connect to the JD Edwards EnterpriseOne server.
11. Click **Finish**.

## CHAPTER 4

# JD Edwards EnterpriseOne Data Objects

This chapter includes the following topics:

- [JD Edwards EnterpriseOne Data Objects Overview , 19](#)
- [JD Edwards EnterpriseOne Data Object Properties, 20](#)
- [JD Edwards EnterpriseOne Data Object Read Operation Properties, 21](#)
- [JD Edwards EnterpriseOne Data Object Write Operation Properties, 25](#)
- [JD Edwards EnterpriseOne Data Object Interface Table Write Operation Properties, 26](#)
- [Importing a JD Edwards EnterpriseOne Data Object, 28](#)
- [Configuring Filters to Import a JD Edwards EnterpriseOne Data Object, 28](#)
- [Editing JD Edwards EnterpriseOne Source Objects, 30](#)
- [Creating a JD Edwards EnterpriseOne Data Object Operation, 30](#)
- [Rules and Guidelines for JD Edwards EnterpriseOne Objects, 30](#)
- [Troubleshooting a Filter, 31](#)

## JD Edwards EnterpriseOne Data Objects Overview

A JD Edwards EnterpriseOne data object is a physical data object that uses a JD Edwards EnterpriseOne table or view as a source or target. A JD Edwards EnterpriseOne data object is the representation of data that is based on a JD Edwards EnterpriseOne table or view.

When you create a data object, you can filter the tables or views you want the Developer Tool to display. After you create a data object, you can edit the data object.

You can configure a data object read, write, or interface table write operation for the data object. Before you configure the operation for the data object, you must establish a connection with JD Edwards EnterpriseOne Enterprise Server. After you configure the operation, you can add the data object read, write, or interface table write operation to a mapping.

The JD Edwards EnterpriseOne data objects represent the tables and views used in the read operation. The tables can represent base tables or interface tables in JD Edwards EnterpriseOne. The tables and views are grouped by system codes. You can add or remove tables and views when you configure the read operation.

You can configure a data object write operation for a JD Edwards EnterpriseOne data object to write data to base and interface tables. You can configure a write operation for small transactional updates. The data

object write operation properties determine how you can load data to JD Edwards EnterpriseOne base and interface tables. If you configure a write operation to write data to the interface tables, you can later push the data to the base tables in JD Edwards EnterpriseOne server with a custom function.

To write data in bulk to JD Edwards EnterpriseOne, you must configure an interface table write operation. When you run a mapping with an interface table write operation, the Data Integration Service connects to a JD Edwards EnterpriseOne database server and loads data into the interface tables. You can invoke the business function to load data from the interface tables to their corresponding base tables. In an interface table write operation, the bulk update occurs in a batch.

## JD Edwards EnterpriseOne Data Object Properties

The JD Edwards EnterpriseOne **Overview** view displays general information about the JD Edwards EnterpriseOne data object and the object properties that apply to the JD Edwards EnterpriseOne resource that you import.

### General Properties

You can configure the following general properties for a JD Edwards EnterpriseOne data object:

- **Name.** Name of the JD Edwards EnterpriseOne data object.
- **Description.** Description of the JD Edwards EnterpriseOne data object.
- **Connection.** Name of the JD Edwards EnterpriseOne connection. Click **Browse** to select a different JD Edwards EnterpriseOne connection.

### Object Properties

You can configure the following general properties and column properties for the JD Edwards EnterpriseOne resource that you add in the data object:

- **Name.** Name of the JD Edwards EnterpriseOne resource.
- **Description.** Description of the JD Edwards EnterpriseOne resource.
- **Native Name.** Name of the JD Edwards EnterpriseOne resource including the entity set that contains the resource.
- **Path Information.** Relative path to the tables in the JD Edwards EnterpriseOne data object. You cannot edit the path information of the object.
- **Column Properties.** Name, data type, precision, scale, primary key, access type, and description of the columns in the JD Edwards EnterpriseOne resource.
- **Key.** Not applicable.
- **Advanced.** The value of the record type. The record type includes Table, View, or InterfaceTable. You cannot edit the Advanced object property.

# JD Edwards EnterpriseOne Data Object Read Operation Properties

The Data Integration Service reads data from a JD Edwards EnterpriseOne server based on the data object read operation properties that you specify.

When you create a data object read operation, the Developer tool creates a Source transformation and an Output transformation. The Source transformation is named after the resource and represents the data that the Data Integration Service reads from the JD Edwards EnterpriseOne resource. Select the Source transformation to view data such as the name, description, physical name, and path information of the JD Edwards EnterpriseOne resource.

The Output transformation represents the data that the Data Integration Service passes into the mapping pipeline. Select the Output transformation to edit the general, ports, sources, query, run-time, and advanced properties.

## Output Properties of a Data Object Read Operation

The Output transformation defines the run-time properties that the Data Integration Service uses to read data from the JD Edwards EnterpriseOne resource.

In the Output transformation, you can also edit the port properties, add sources, query using filter or sort, and select a different JD Edwards EnterpriseOne connection.

### Ports Properties

The ports properties list the name, data type, precision, scale, and description for all the ports that the data object read operation contains.

You can configure the following ports properties in the data object read operation:

Property	Description
Name	Name of the port.
Type	Data type of the port.
Precision	Maximum number of digits for numeric data types or maximum number of characters for string data types. The precision includes the scale for numeric data types.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the port.

### Source Properties

The source properties display the table or view that the data object read operation contains. You can add or remove the tables or views within the configured read operation. Each table or view is represented by a code.

### Query Properties

The query properties display the filters that the data object read operation contains. You can edit the filter condition.

The Query properties allow you to specify Select Distinct or All, Join, Filter, and Sort conditions in the data object read operation.

### Select Distinct

Configure the Data Integration Service to extract distinct rows from a single JD Edwards EnterpriseOne source. You can configure one of the following values:

- **Distinct.** Selects unique rows from a single JD Edwards EnterpriseOne source. The Data Integration Service includes a `SELECT DISTINCT` statement if you choose this option and filters out duplicate data when you use the JD Edwards EnterpriseOne data object in a mapping.
- **All.** Extracts duplicate rows.

### Join

Specify the join type to join two or more JD Edwards EnterpriseOne sources. Join enables only when you have two or more tables. 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.

### Filter

Use a filter to reduce the number of rows that the Data Integration Service reads from the source. The filter specifies the `where` clause of select statement. When you enter a source filter, the Developer tool adds a `WHERE` clause to the default query. You can use the native or platform following expression to select specific records.

The following table describes the Platform Expression properties you specify when you filter records from JD Edwards EnterpriseOne tables or views:

Property	Description
Left Field	The column of the JD Edwards EnterpriseOne object on which you want to apply the filter condition.
Operator	Simple operators you can use to filter records. You can select one of the following operators:  =, !=, <, <=, >, >=
Right Field	The literal value you specify to filter the JD Edwards EnterpriseOne objects.

To specify a native expression, use the following syntax:

- 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. You can use a period (.) to separate the table name and column name.

The following table describes the operators that you can use in Native and Platform Expression filters:

Operator	Expression	Description
<	Platform, Native	Extracts data where value of a field is lesser than the value of a literal or the other field. For example, F0101.AN8 < 100.
>	Platform, Native	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.
=	Platform, Native	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.
<=	Platform, Native	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.
>=	Platform, Native	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.
!=	Platform, Native	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	Native	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	Native	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	Native	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.

Operator	Expression	Description
BETWEEN..AND, between..and	Native	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	Native	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
IN, in	Native	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	Native	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)

### Sort

You can sort the rows queried from JD Edwards EnterpriseOne source. The Data Integration Service adds the ports to the `ORDER BY` clause in the default query. You can sort rows extracted from the multiple sources in ascending or descending order. If you do not enter any value, the Data Integration Service does not sort data.

## Run-time Properties

The run-time properties display the name of the connection that the Data Integration Service uses to read data from the JD Edwards EnterpriseOne resource. You can select a different connection or parameterize the connection.

## Advanced Properties

The advanced properties display the operation type for the Data Integration Service.

## 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.
- Enclose string values in single quotes.



- Enter the literal value based on the data type of the column specified in the first token. For example,  

```
F0101.AN8 < 100
F0101.ALPH LIKE 'MARKETING COMPANY'
```
- Specify the literal value in the YYYY-MM-DD format if the data type is Jdedate. For example,  

```
F0101.UPMJ = 2003-05-29
```
- Specify the literal value in the YYYY-MM-DD HH:MM:SS format if the data type is Jdeutime. For example,  

```
F01301.TDSTR = 2003-05-29 15:10:25
```
- Specify the literal value in the HH:MM:SS format if the data type 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.
- You cannot apply source filter on a binary field.

## JD Edwards EnterpriseOne Data Object Write Operation Properties

The Data Integration Service writes data to a JD Edwards EnterpriseOne Data table or view based on the data object write operation properties that you specify.

When you create a data object write operation, the Developer tool creates an Input transformation and a Target transformation.

The Input transformation defines the run-time properties that the Data Integration Service uses to write data to the tables or views in JD Edwards EnterpriseOne. In the Input transformation, you can also edit the port and target properties. You can also edit the run-time properties to select a different JD Edwards EnterpriseOne connection.

### Input Properties of a JD Edwards EnterpriseOne Data Object Write Operation

The Input transformation defines the run-time properties that the Data Integration Service uses to write data to the JD Edwards EnterpriseOne table or views.

In the Input transformation, you can also edit the port properties and select a different JD Edwards EnterpriseOne connection.

### Target Properties of a JD Edwards EnterpriseOne Data Object Write Operation

The target properties represent the data that the Developer tool populates based on the JD Edwards EnterpriseOne table that you added when you created the data object. You can view the target properties of the data object write operation from the **General**, **Column**, and **Advanced** tabs.

## Ports Properties

The ports properties list the name, data type, precision, scale, and description for all the ports that the data object write operation contains.

You can configure the following ports properties in the data object write operation:

Property	Description
Name	Name of the port.
Type	Data type of the port.
Precision	Maximum number of digits for numeric data types or maximum number of characters for string data types. The precision includes the scale for numeric data types.
Scale	Maximum number of digits after the decimal point for numeric values.
Description	Description of the port.

## Target Properties

The target properties display the JD Edwards EnterpriseOne tables that the Data Integration Service writes data to.. You can remove the table and add a different table to which you want to write data to.

## Run-time Properties

The run-time properties display the name of the connection that the Data Integration Service uses to write data to the JD Edwards EnterpriseOne table. You can select a different connection or parameterize the connection.

# JD Edwards EnterpriseOne Data Object Interface Table Write Operation Properties

When you configure an interface table write operation, the Data Integration Service writes data to a JD Edwards EnterpriseOne Data interface table based on the data object interface table write operation properties that you specify.

When you create a data object interface table write operation, the Developer tool creates an Input transformation and a Target transformation.

The Input transformation defines the advanced properties that the Data Integration Service uses to write data to the interface tables in JD Edwards EnterpriseOne. Select the invoke business function if you want to start the batch process to load data from interface tables to the base tables in JD Edwards EnterpriseOne.

In the Input transformation, you can also edit the port and target properties. You can also edit the run-time properties to select a different JD Edwards EnterpriseOne connection.

# Input Properties of a JD Edwards EnterpriseOne Data Object Interface Table Write Operation

The Input transformation defines the run-time properties that the Data Integration Service uses to write data in bulk to the JD Edwards EnterpriseOne interface table.

In the Input transformation, you can edit the general and port properties. You can add or remove an interface table from the target properties. You can add a new JD Edwards EnterpriseOne connection in the run-time properties and you can configure advanced properties.

## Advanced Properties

The advanced properties determine how the Data Integration Service writes data to the JD Edwards EnterpriseOne interface table.

You can configure the following advanced properties in the data object interface table write operation:

### Table Prefix

The database and the schema name of the interface table in JD Edwards EnterpriseOne where you want to write the data. You can use a parameter for this attribute.

Specify the table name prefix in the following format:

`database_name.owner`

For example, in Microsoft SQL Server, specify the table prefix as: *JDE\_DEVELOPMENT.TESTDTA*

Where:

- *JDE\_DEVELOPMENT* is the database.
- *TESTDTA* is the schema name.

To specify a prefix for multiple interface tables, separate each prefix by a comma.

### Truncate Table

The Data Integration Service truncates the target interface table before it begins to load the data. Default is disabled.

You must select **Truncate Table** to truncate the interface table before the Data Integration Service loads data into the interface table.

### Invoke Business Function

Executes the business function that invokes the batch process. The batch process extracts data from interface tables and writes the data to base tables. Select this option if you want to write the data from the interface tables to the base tables.

### Input Parameter File Path

The path of the input parameter file on the machine where the Data Integration Service runs. The Data Integration Service reads the contents of the input parameter file to determine which base table to write the data to.

### Job Status File Path

The file path on the local machine where the status code of the invoked business function is logged.

# Importing a JD Edwards EnterpriseOne Data Object

Import a JD Edwards EnterpriseOne data object to specify the JD Edwards EnterpriseOne resource from which you want to read data. You can then create a data object read operation based on the data object.

1. Select a project or folder in the **Object Explorer** view.
  2. Click **File > New > Data Object**.
  3. Select **JD Edwards EnterpriseOne Data Object** and click **Next**.
- The **New JD Edwards EnterpriseOne Data Object** dialog box appears.
4. Enter a name for the data object.
  5. Click **Browse** next to the **Location** option and select the target project or folder.
  6. Click **Browse** next to the **Connection** option and select the JD Edwards EnterpriseOne connection from which you want to import the JD Edwards EnterpriseOne URL metadata.
  7. To add a resource from the JD Edwards EnterpriseOne URL, click **Add** next to the **Selected Resources** option.

The **Add Resource** dialog box appears.

8. Select a table or view. You can search or navigate to the table or view.
  - Navigate to the table or view that you want to import, and click **OK**.
  - To search for a table or view, enter the name of the table or view that you want to add in a basic filter, and click **OK**.
  - To search for a table or view, enter the name, description, or the record type filter in an advanced filter, and click **OK**.
9. If required, add more tables and views to the JD Edwards EnterpriseOne data object.

You can also add tables and views to a JD Edwards EnterpriseOne data object after you create it.
10. Click **Finish**.

The data object appears under Physical Data Objects in the project or folder in the **Object Explorer** view.

## Configuring Filters to Import a JD Edwards EnterpriseOne Data Object

When you connect to JD Edwards EnterpriseOne to import a JD Edwards EnterpriseOne data object, the Developer Tool displays tables and views that you can import. The tables and views are grouped by system codes.

You can enter a filter to reduce the number of tables or views that appear while you import. If you do not enter a filter, all the tables and views appear. You can specify a simple filter or an advanced filter. When you use the advanced filter, you can also specify the record type that you want to filter.

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 you can use 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 you can use 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 you can use 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 you can use 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.

# Editing JD Edwards EnterpriseOne Source Objects

After you import JD Edwards EnterpriseOne source objects, you can edit the source object to modify the name, native type, data type, precision, key type, and description fields. You can change the scale only for MATH\_NUMERIC data types.

## Working with the Jdetime Data Type

When you import a source object, the Developer Tool imports columns of the Jdetime data type as Math\_Numeric. To process the data correctly during a mapping, you must edit the source object and manually change the data type to Jdetime. Edit the data type of a column on the **Columns** tab of a source object.

# Creating a JD Edwards EnterpriseOne Data Object Operation

Create a JD Edwards EnterpriseOne data object read, write, or interface table write operation from a JD Edwards EnterpriseOne data object. You can then add the operation as a source in a mapping.

1. Select the JD Edwards EnterpriseOne data object in the **Object Explorer** view.
2. Right-click the data object and select **New > Data Object Operation**.  
The **Data Object Operation** dialog box appears.
3. Enter a name for the data object operation.
4. Select **Read**, **Write**, or **InterfaceTable Write** as the type of data object operation.
5. Click **Add**.  
The **Select Resources** dialog box appears.
6. Select the JD Edwards EnterpriseOne resource for which you want to create the data object operation, and click **OK**.
7. Click **Finish**.

The Developer tool creates the data object operation for the selected JD Edwards EnterpriseOne data object.

# Rules and Guidelines for JD Edwards EnterpriseOne Objects

Consider the following rules and guidelines for JD Edwards EnterpriseOne objects:

- You cannot use the JOIN condition on JD Edwards EnterpriseOne views because of a restriction from the JDBj APIs.
- The platform expression does not support the Jdetime, Jdedate, and Jdeutime data types.
- When you configure a query for a data object read operation, you cannot configure Distinct with a Join query property. Also, do not specify the Join query along with the filter query. The Data Integration Service incorrectly reads data because of a restriction from the JDBj APIs.

- You cannot use pushdown optimization when you use data filters with expressions that contain the OR or to\_date conditions.

## Troubleshooting a Filter

**I get the following error message when I use a native filter to read data from a JD Edwards EnterpriseOne table "The Table Name provided in the native filter condition is incorrect. Provide a valid table name."**

The error occurs because the filter that you specified to read data from a JD Edwards EnterpriseOne table contains a period in the literal part of the filter condition. The Data Integration Service treats the period as `<Tablename.columnname> <operator> <Tablename.columnname>` of the syntax.

For example, you encounter an error for the following filter conditions:

- `(F01301.TDSTR=2019-02-19 00:00:00)`
- `F01301.TDSTR=2019-01-01 00:00:00.000`

## CHAPTER 5

# JD Edwards EnterpriseOne Mappings

This chapter includes the following topic:

- [JD Edwards EnterpriseOne Mappings Overview, 32](#)

## JD Edwards EnterpriseOne Mappings Overview

After you create a JD Edwards EnterpriseOne data object read, write, or interface table write operation, you can develop a mapping.

You can define the following objects in the mapping:

- JD Edwards EnterpriseOne data object read operation as the input to read data from JD Edwards EnterpriseOne metadata.
- JD Edwards EnterpriseOne data object write operation as the output to write data to JD Edwards EnterpriseOne.
- JD Edwards EnterpriseOne data object interface table write operation as the output to write data to JD Edwards EnterpriseOne interface tables and subsequently to base tables.

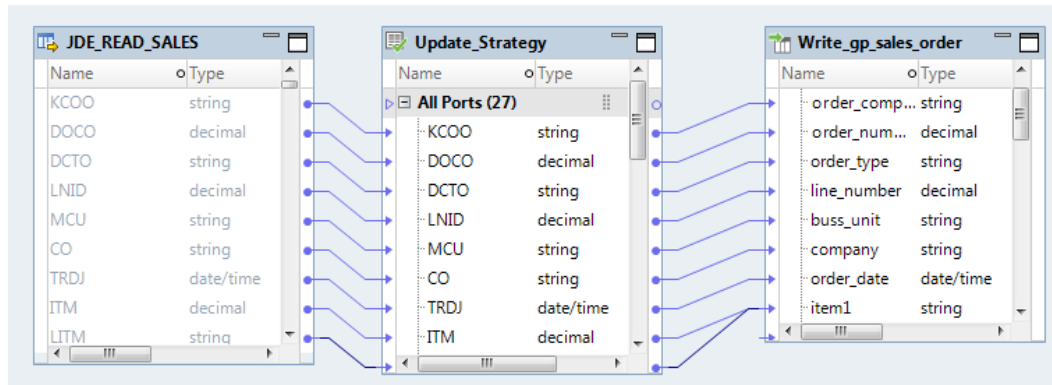
Validate and run the mapping to read data and write data to JD Edwards EnterpriseOne server.



## JD Edwards EnterpriseOne Mapping Read Example

You are a data administrator in a product organization. You want to migrate unique sales order details from JD Edwards EnterpriseOne to Greenplum for analysis.

The following image shows the JD Edwards EnterpriseOne mapping example:



The mapping contains a *JDE\_READ\_SALES* read operation to read the sales information for your organization. The expression transformation is used to calculate the *net\_quantity* and *net\_amount* for each sales order. The mapping contains a *Write\_gp\_sales\_order* write operation to which you want to insert rows to the Greenplum table.

### Mapping Input

The data object for the mapping is a JD Edwards EnterpriseOne table *F42119* that contains the sales order history. The source contains information for the sales order details, such as the *company name*, *order type*, *quantity*, *unit price*, and *order date*.

### Transformations

Add the expression transformation to calculate the *net\_quantity* and *net\_amount* for each sales order. Load the data to the Greenplum table with information related to each sales transaction, such as the *company\_name*, *order\_date*, and *item\_purchased*.

### Mapping Output

The data object for the mapping is a Greenplum table. To write data to the Greenplum table, create a data object *gp\_sales\_order*. Some of the target columns include the *company name*, *order type*, *quantity*, *unit price*, *order date*, *net\_quantity*, and *net\_amount*.

When you run the mapping, the Data Integration Service loads the sales information to the Greenplum table.

The following image shows the sales information that the Data Integration Service writes to the Greenplum table:

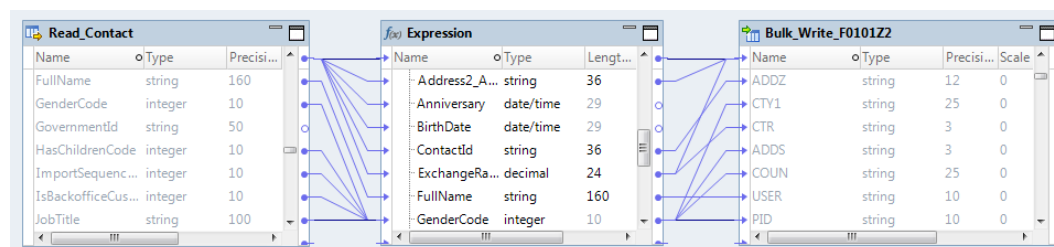
	Name	Native Type	Precisi...	Scale	Primar...	Nullable	Description
1	order_company	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	order_number	decimal	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	order_type	varchar	50	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	line_number	decimal	10	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	buss_unit	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	company	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	order_date	date	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	item1	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	item2	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	item3	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	location	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
12	description	varchar	500	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
13	sales_category	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
14	sub_section	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
15	quantity	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
16	quantity_shipped	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
17	unit_price	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
18	unit_cost	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
19	weight	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
20	currency_code	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
21	exchange_rate	decimal	10	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
22	transaction	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
23	user_id	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
24	program_id	varchar	50	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Analysts can run queries and generate reports based on the data in the Greenplum table.

## JD Edwards EnterpriseOne Mapping Interface Table Write Example

Your organization has a presales team that manages user accounts in an enterprise resource system such as Microsoft Dynamics. The sales team manages data in JD Edward EnterpriseOne. You want to migrate all the account information from Microsoft Dynamics to JD Edwards EnterpriseOne.

The following image shows the JD Edwards EnterpriseOne mapping example:



The mapping contains a Microsoft Dynamics source that contains the contact information of account holders. The Expression transformation concatenates the first name and last name to provide the full name

of the account holder. The mapping output contains the JD Edwards EnterpriseOne data object to write data in bulk to the JD Edwards EnterpriseOne table.

## Mapping Input

The source object for the mapping is a Microsoft Dynamics table *Contact*. The source contains the contact information, such as user ID, mailing name, city country, and workstation ID. To read data from the JD Edwards EnterpriseOne table, create a data object read operation *Read\_Contact*.

## Transformations

You can add the Expression transformation to concatenate the first name and last name to obtain the full name of the account holder. Use the Expression transformation to change the data/time format of Microsoft Dynamics to a compatible format of JD Edwards EnterpriseOne.

## Mapping Output

The mapping output is a JD Edwards EnterpriseOne data object. To write data to the JD Edwards EnterpriseOne table *F0101Z2*, create a data object interface table write operation *Bulk\_Write\_F0101Z2*.

Some of the target columns include the user ID, mailing name, city country, workstation ID. When you run the mapping, the Data Integration Service loads the contact information to the JD Edwards EnterpriseOne table.

The following image shows the contact information that the Data Integration Service writes to the JD Edwards EnterpriseOne table:

	Name	Native Name	Type	Precision	Scale	Primary Key	Access Type	Description
1	EDUS	EDUS	STRING	10	0	<input checked="" type="checkbox"/>	Read and Write	User ID
2	EDBT	EDBT	STRING	15	0	<input checked="" type="checkbox"/>	Read and Write	Batch Number
3	EDTN	EDTN	STRING	22	0	<input checked="" type="checkbox"/>	Read and Write	Transaction Nu...
4	EDLN	EDLN	MATH_NU...	7	3	<input checked="" type="checkbox"/>	Read and Write	Line Number
5	MLNM	MLNM	STRING	40	0	<input type="checkbox"/>	Read and Write	Mailing Name
6	ADD1	ADD1	STRING	40	0	<input type="checkbox"/>	Read and Write	Address Line 1
7	ADD2	ADD2	STRING	40	0	<input type="checkbox"/>	Read and Write	Address Line 2
8	ADD3	ADD3	STRING	40	0	<input type="checkbox"/>	Read and Write	Address Line 3
9	ADD4	ADD4	STRING	40	0	<input type="checkbox"/>	Read and Write	Address Line 4
10	ADDZ	ADDZ	STRING	12	0	<input type="checkbox"/>	Read and Write	Postal Code
11	CTY1	CTY1	STRING	25	0	<input type="checkbox"/>	Read and Write	City
12	CTR	CTR	STRING	3	0	<input type="checkbox"/>	Read and Write	Country
13	ADDS	ADDS	STRING	3	0	<input type="checkbox"/>	Read and Write	State
14	COUN	COUN	STRING	25	0	<input type="checkbox"/>	Read and Write	County
15	USER	USER	STRING	10	0	<input type="checkbox"/>	Read and Write	User ID
16	PID	PID	STRING	10	0	<input type="checkbox"/>	Read and Write	Program ID
17	JOBID	JOBID	STRING	10	0	<input type="checkbox"/>	Read and Write	Work Station ID
18	UPMJ	UPMJ	JDEDATE	6	0	<input type="checkbox"/>	Read and Write	Date Updated

# APPENDIX A

## Data Type Reference

This appendix includes the following topic:

- [JD Edwards EnterpriseOne and Transformation Data Types, 36](#)

## JD Edwards EnterpriseOne and Transformation Data Types

PowerCenterInformatica 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 ServiceData Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration ServiceData 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 ServiceData 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 PowerCenterInformatica 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

JD Edwards EnterpriseOne Data Type	Description	Transformation Data Type	Description
Char	Precision 1	Text	1 to 65,000 characters
Int	Precision 10	Integer	-2,147,483,648 to 2,147,483,647
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)
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	Datetime	date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to nanosecond)
Math_Numeric	Precision 1 to 28 digits, scale 0 to 28	Decimal	Precision 1 to 28 digits, scale 0 to 28
String	1 to 65,000 characters	String	1 to 65,000 characters
Varstring	1 to 65,000 characters	String	1 to 65,000 characters

## Jdetime Data Type

PowerCenterInformatica 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. When you import a data object that uses a Jdetime data type, the Developer Tool imports the data object 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. The Data Integration Service does not import every column of the Math\_Numeric data type from the Jdetime data type. Before you change the data type of a Math\_Numeric column, verify that the JD Edwards EnterpriseOne data object uses a Jdetime data type.

# INDEX

## A

architecture  
Data Integration Service and JD Edwards EnterpriseOne integration [9](#)

## B

backslash character  
regular expressions for PowerExchange for JD Edwards EnterpriseOne [28](#)

## C

configuration  
  .ini files, copying [11](#)  
  API libraries, copying [11](#)  
  database driver libraries, copying [11](#)  
creating  
  JD Edwards EnterpriseOne connection [17](#)

## D

data types  
  JD Edwards EnterpriseOne data types [36](#)  
  Jdetime for JD Edwards EnterpriseOne [37](#)  
  PowerExchange for JD Edwards EnterpriseOne [36](#)  
  transformation data types for JD Edwards EnterpriseOne [36](#)

## F

filters  
  regular expressions for PowerExchange for JD Edwards EnterpriseOne [28](#)  
  SQL expressions for PowerExchange for JD Edwards EnterpriseOne [28](#)  
  valid filter conditions for PowerExchange for JD Edwards EnterpriseOne [28](#)

## J

JD Edwards [7](#)  
JD Edwards EnterpriseOne  
  overview [19](#)  
JD Edwards EnterpriseOne connection  
  creating [17](#)  
  overview [15](#)  
  properties [16](#)  
JD Edwards EnterpriseOne data objects  
  importing [28](#)  
  properties [20](#)

JD Edwards EnterpriseOne data types  
  Jdetime [30](#)  
JD Edwards EnterpriseOne Enterprise Server  
  data integration [9](#)  
JD Edwards EnterpriseOne environment  
  Data Integration Service, connecting [9](#)  
JD Edwards EnterpriseOne interface table write operation  
  output transformation [21](#), [25](#), [26](#)  
  target transformation [25](#), [26](#)  
JD Edwards EnterpriseOne mapping interface table write example [34](#)  
JD Edwards EnterpriseOne mapping read example [33](#)  
JD Edwards EnterpriseOne operation  
  creating [30](#)  
JD Edwards EnterpriseOne read operation  
  output transformation [21](#), [25](#), [26](#)  
  source transformation [21](#)  
JD Edwards EnterpriseOne read operation properties  
  advanced [24](#)  
  ports [21](#), [26](#)  
  query [21](#)  
  run-time [24](#)  
  source [21](#)  
JD Edwards EnterpriseOne write operation  
  output transformation [21](#), [25](#), [26](#)  
  target transformation [25](#), [26](#)  
JD Edwards EnterpriseOne write operation properties  
  ports [21](#), [26](#)

## L

LDAP  
  mapping overview [32](#)

## M

mapping interface table write example [34](#)  
mapping read example [33](#)  
metadata  
  JD Edwards EnterpriseOne tables and views [8](#)

## P

PowerExchange for JD Edwards EnterpriseOne  
  source objects [28](#)  
PowerExchange for JD Edwards EnterpriseOne source objects  
  filters, working [28](#)  
  Jdetime data type, working [30](#)

## R

### regular expressions

- backslash character for PowerExchange for JD Edwards EnterpriseOne [28](#)
- valid filter conditions for PowerExchange for JD Edwards EnterpriseOne [28](#)

## S

### system codes

- description [7](#)
- JD Edwards EnterpriseOne tables [8](#)
- JD Edwards EnterpriseOne views [8](#)

## T

### tables

- JD Edwards EnterpriseOne metadata [8](#)
- JD Edwards EnterpriseOne system codes [8](#)

## V

### views

- JD Edwards EnterpriseOne metadata [8](#)
- JD Edwards EnterpriseOne system codes [8](#)