SAP BusinessObjects™

Supplement for J.D. Edwards

- SAP BusinessObjects Data Services 4.1 Support Package 1 (14.1.1.0)

2012-12-12

SAP

2012-12-12

Contents

Overview

The SAP BusinessObjects Data Services J.D. Edwards (JDE) interface is a license-controlled feature. With this interface, you can use the Designer to:

- View modules and tables in a database instance under a specified J.D. Edwards environment
- Import metadata for tables into SAP BusinessObjects Data Services
- Create batch data flows that use J.D. Edwards tables as sources

Related Topics

- System requirements
- Datastores

System requirements

The J.D. Edwards interface allows you to connect SAP BusinessObjects Data Services with two J.D. Edwards applications:

- World
- OneWorld

Both of these applications operate on an underlying database. Install the drivers you need to connect J.D. Edwards with SAP BusinessObjects Data Services on the same computers on which you install the Designer and Job Server components.

2.1 World

The J.D. Edwards interface supports World version A7.3 and later. SAP BusinessObjects Data Services only supports this application when World uses an underlying DB2 database on an AS/400 computer. Use either an ODBC datastore connection (using IBM's iSeries Access driver) or a Detail_DB2 datastore connection.

Related Topics

- Defining a J.D. Edwards World datastore

2.2 OneWorld

The J.D. Edwards interface supports OneWorld version B7.3 and compatible versions, including Xe. SAP BusinessObjects Data Services supports this application if OneWorld uses one of three underlying databases:

- DB2 for AS/400 (use an ODBC datastore connection using IBM's iSeries Access)
- Microsoft SQL Server
- Oracle

J.D. Edwards OneWorld environment data sources include system-control data sources and business data sources. Your J.D. Edwards system-control data sources and your business data sources must reside on the same database instance. If a system control table is not located in the same database instance, you cannot create a datastore for that J.D. Edwards application.

Related Topics

- Defining a J.D. Edwards OneWorld datastore

Datastores

SAP BusinessObjects Data Services uses datastore connections to link with other applications or databases. In a design environment, you use datastores to browse, search, or import metadata that represents external tables, files, messages, and other database objects. When running jobs, the software uses datastore information to move data between source and target databases and applications.

After defining a J.D. Edwards datastore in SAP BusinessObjects Data Services, you can browse application modules, tables, table descriptions, and column descriptions.

You must define J.D. Edwards datastore connections with accurate J.D. Edwards information to ensure the accessibility of the tables.

3.1 Defining a J.D. Edwards World datastore

You can use two methods to access the J.D. Edwards World application:

- ODBC driver
- Mainframe interfaces

Both options access a DB2 for AS/400 database.

3.1.1 ODBC driver

You can access the J.D. Edwards World application using an ODBC driver called iSeries Access. Before creating a datastore using this ODBC driver, you must install and configure the driver on the same computers on which you installed the Designer and Job Server components. Use the ODBC Administration utility to install and configure the driver. In the driver's library list, be sure to list all the libraries needed in your particular J.D. Edwards environment.

3.1.2 Mainframe interfaces

You can access the J.D. Edwards World application using Mainframe interfaces. Refer to the pertinent documentation for installation and setup instructions.

3.1.2.1 To define a J.D. Edwards World datastore

1. In the object library of the Designer, go to the **Datastores** tab.
2. Right-click inside the object library window and select **New**.
3. In the Datastore Editor window, enter a name for this datastore (`DS_JDEWorld`, for example).
4. In the **Application type** list, select **JDE World**.
5. In the **Database Type** list, select **ODBC**.
6. Enter the connection information.
7. Click the **JDE World Properties** tab and enter the required information. SAP BusinessObjects Data Services handles J.D. Edwards data in a logical grouping of libraries.

Option	Description
Environment	Enter the J.D. Edwards application environment name.
Local library	Enter the name of the library where SAP BusinessObjects Data Services can find the J.D. Edwards system control table F0005.
Data dictionary library	Enter the name of the library where SAP BusinessObjects Data Services can find the J.D. Edwards data dictionary table F9201.
Security library	Enter the name of the library where SAP BusinessObjects Data Services can find J.D. Edwards security tables, such as F0094.

8. Click **OK**.

   The J.D. Edwards datastore appears in the object library.

Related Topics

- Reference Guide: Objects, Database datastores

3.2 Defining a J.D. Edwards OneWorld datastore

A J.D. Edwards environment uses a logical construct called a data source to associate the J.D. Edwards application layer data with the database layer data (for example, Microsoft SQL Server). J.D. Edwards uses some data sources to associate system-control data and uses other data sources to associate business data.

The distributed nature of J.D. Edwards architecture allows an instance of J.D Edwards to have data sources on multiple database servers. For example, a finance data source can be on a Microsoft SQL Server while the system-control data source is on Oracle. Because a SAP BusinessObjects Data Services datastore can only be associated with one database server, you must follow these rules when defining a J.D. Edwards OneWorld datastore connection in SAP BusinessObjects Data Services:

- The data sources within a J.D. Edwards environment that you want to access must be in the same Microsoft SQL Server, DB2 for AS/400, or Oracle instance.
- You must specify required system control data sources, and in some cases, you must enter the owner ID of the data source. For Microsoft SQL Server, you must also enter the database name for each, and for DB2 for AS/400, you must enter the library for each.
- If using Microsoft SQL Server, you must configure a new datastore for each business data source on a unique database even if all databases are on the same Microsoft SQL Server. For example, if you want to extract Human Resource and Finance information from separate databases on the same Microsoft SQL Server instance, you must create two different datastores.

3.2.1 To define a J.D. Edwards OneWorld datastore

1. In the object library of the Designer, go to the **Datastores** tab.
2. Right-click inside the object library window and select **New**.
3. In the Datastore Editor window, enter a name for this datastore (`DS_JDEOneWorld`, for example).
4. In the **Application Type** list, select **JDE_One_World**.
5. In the **Database Type** list, select the database on which your J.D. Edwards application runs.

If you choose:	Do this:
Microsoft SQL Server	Enter the database-related information including the user name and password for the database. In the **Database server name** box, enter the name of the SQL Server instance. In the **Database name** box, enter the name of the database containing your business data, such as a finance database. If you have more than one database of business data create a separate J.D. Edwards datastore like the first except enter a different database name here. **Note:** With Microsoft SQL Server as your database, the Job Server must be installed on Windows NT or 2000.
Oracle	Enter the information required to connect to the Oracle database.

6. Click the **JDE OneWorld Properties** tab and enter the required information: the environment name and the databases to which the data sources point.

   Some J.D. Edwards data sources contain system-control information. SAP BusinessObjects Data Services needs this system-control information to properly translate J.D. Edwards data. Like all data sources, these system-control data sources can be anywhere on the Microsoft SQL Server, DB2 for AS/400, or Oracle instance. Enter the database names for each data source.

Option	Description
Environment	Enter the J.D. Edwards application environment name.
System data source	Enter the name of the database where the tables F986101, F98611, and F00941 are located. This option is available for DB2 and Microsoft SQL Server databases.
System data source owner	Enter the owner ID for the system data source. This option is available for Microsoft SQL Server and Oracle databases.
Object librarian data source	Enter the name of the database where the tables F9860 and F9861 are located. This option is available for DB2 and Microsoft SQL Server databases.

Option	Description
Local data source	Enter the name of the database where the table F0005 is located. This option is available for DB2 and Microsoft SQL Server databases.
Data dictionary data source	Enter the name of the database where the table F9203 is located. This option is available for DB2 and Microsoft SQL Server databases.

7. Click **OK**.
   The J.D. Edwards OneWorld datastore appears in the object library.

3.3 Browsing and importing metadata

After creating a J.D. Edwards datastore, you can browse and import the metadata from the connected application.

3.3.1 To browse and import the metadata

1. View the modules in the J.D. Edwards application. You can:
   - Right-click the datastore name and select **Open**
   - Double-click the datastore name

   The workspace shows the list of application modules.
2. View the tables in a particular module. You can:
   - Expand the module tree
   - Double-click a module (folder)
3. To import table data, right-click a table name and select **Import**.

   Note:
   When you import a table into SAP Data Services via the J.D. Edwards interface, the software does not preserve the table hierarchy. All tables are listed at the same level.

You can also import tables using the Designer**Import by Name** and **Search** features.

3.4 Extracting data from J.D. Edwards systems

When extracting data from either J.D. Edwards application, SAP BusinessObjects Data Services processes and converts data types appropriately. For example, the software recognizes null values for date columns. J.D. Edwards, on the other hand, does not support null values. Instead, J.D. Edwards stores dates as numeric values. When date data is not present, J.D. Edwards stores the number 0. Therefore, the software automatically translates a value of zero for a date to NULL. If you expect a column to have zeros in it, do not use the date type in the primary key.

SAP BusinessObjects Data Services automatically translates most data types. In some cases, you must translate output explicitly.

Related Topics

- Data types

Reference information

4.1 Data types

SAP BusinessObjects Data Services uses unique processing to translate some J.D. Edwards data types:

- Translating decimals
- Translating dates
- Translating time

4.1.1 Translating decimals

J.D. Edwards translates decimal values using an additional piece of information from the underlying database: the decimal separator shift integer. This value indicates the number of digits to move left from the end of the value number. For example, to represent a number like 1.23, J.D. Edwards stores the value as "123" with a decimal separator shift of "2" (starting at the end of the number, the decimal shifts two places to the left).

To process J.D. Edwards decimal values, the software extracts the numeric value stored in the underlying database and then applies the decimal separator shift to determine where the decimal belongs. The software applies the translation when a decimal column appears in a SQL statement against the database.

SAP BusinessObjects Data Services interprets J.D. Edwards currency data types as decimals.

The software does not automatically translate decimal data types in three cases:

- SQL transform
- sql function
- pushdown_sql function

In these cases, translate decimals explicitly. To translate decimals explicitly, you must know the decimal shift value for your selected column. Use the shift number to determine the denominator for decimal

translation. For example, if column `COL26` in table `JDETAB` is a JDE decimal column and it has a shift value of `2`, you would manually modify the SQL statement to say something like `SELECT COL26/100 FROM JDETAB`. The denominator in this statement has two zeros following the `1`. If the shift value was `3`, the statement would be `SELECT COL26/1000 FROM JDETAB`.

4.1.2 Translating dates

J.D. Edwards stores dates as numeric values. The J.D. Edwards date format is similar to Julian date format except the year starts with 1900.

SAP BusinessObjects Data Services handles J.D. Edwards numeric values by reading the date information and translating with an internal formula based on the J.D. Edwards start year. For example, the software would handle the date February 18, 1999 as follows:

Year = (1999 - 1900) * 1000;

Day = 31 + 18; [31 is the # of days in January]

The resulting Day value is the number of days since 1/1/1999. The Julian date value is Year + Day = `99049`.

Like decimal translation, there are three cases when the software does not automatically translate dates:

- SQL transform
- sql function
- pushdown_sql function

In these cases, you must use the `JDE_Date` function. See JDE_Date for details.

4.1.3 Translating time

SAP BusinessObjects Data Services does not automatically interpret J.D. Edwards time data types. If you need to translate a number to its time value, you can use the `JDE_Time` function.

Related Topics

- JDE_Time

4.2 Functions

When conversion is not automatic, you can use SAP BusinessObjects Data Services functions to convert J.D. Edwards data to internal SAP BusinessObjects Data Services data types:

- JDE_Date
- JDE_Time

You can access these functions from the query editor: go to the **Mapping** or the **Where** tabs, and click the **Functions** button.

Note:
You cannot use the lookup function with a J. D. Edwards datastore. Use the lookup_ext function instead.

4.2.1 JDE_Date

Given a Julian date stored in a J.D. Edwards database, `JDE_Date` returns the equivalent value as a SAP BusinessObjects Data Services date data type.

Syntax

```
JDE_Date(jde_julian_date)
```

Return Value

date: The SAP BusinessObjects Data Services date data type equivalent to the specified Julian date.

Where

`jde_julian_date` : The integer column in the input table that contains a J.D. Edwards Julian date.

Example:

Function	Results
`JDE_Date (99049)`	`2/18/1999`

You can translate J.D. Edwards Julian dates using column mapping. When you input a J.D. Edwards Julian date number (for example, `99049`), an SAP BusinessObjects Data Services date type results (for example, `2/18/1999`). Enter the function in the **Mapping** section of the Query transform editor.

Use this function when the software does not automatically translate dates.

Note:
Because this function takes an integer as input, map the incoming column to an integer column before applying the function.

Related Topics

- Translating dates

4.2.2 JDE_Time

Given a number representing time in J.D. Edwards, `JDE_Time` returns an SAP BusinessObjects Data Services character value that represents the equivalent military time in *`HH24:MI:SS`* format (where *`HH`* is hours, *`MI`* is minutes, and *`SS`* is seconds).

Syntax

```
JDE_Time(time_integer)
```

Return Value

Char(8): The time in *`HH24`*: *`MI`* : *`SS`* format.

Where

`time_integer` : An integer column in the input table that represents a J.D. Edwards time.

Example:

Function	Results
`JDE_Time(92513)`	`09:25:13`

You can translate times from J.D. Edwards formatting into SAP BusinessObjects Data Services formatting using column mapping. For instance, if you input a J.D. Edwards record update time of 92513, the resulting SAP BusinessObjects Data Services time would be 9:25:13. Enter the function in the **Mapping** section of the Query transform editor.

Index