JD Edwards EnterpriseOne Tools

Development Tools: Data Access Tools Guide

Release 8.98 Update 4

E14702-02

March 2011

JD Edwards EnterpriseOne Tools Development Tools: Data Access Tools Guide, Release 8.98 Update 4

E14702-02

Contents

6 Understanding Business Views

7 Designing Business Views

8 Understanding Table Conversion

9 Setting Up Table Conversions

10 Running Table Conversions

11 Preparing Foreign Tables for Table Conversion

Glossary

Index

Preface

Welcome to the JD Edwards EnterpriseOne Tools Data Access Toolst Guide.

Audience

This guide is intended for system administrators and technical consultants who are responsible for assembling, building, and deploying packages.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Object Management Workbench and EnterpriseOne objects.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/us/corporate/accessibility/index.html`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/support/contact.html` or visit `http://www.oracle.com/accessibility/support.html` if you are hearing impaired.

Related Documents

You can access related documents from the JD Edwards EnterpriseOne Release Documentation Overview pages on My Oracle Support. Access the main documentation overview page by searching for the document ID, which is 876932.1, or by using this link:

`https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=876932.1`

To navigate to this page from the My Oracle Support home page, click the Knowledge tab, and then click the Tools and Training menu, JD Edwards EnterpriseOne, Welcome Center, Release Information Overview.

This guide contains references to server configuration settings that JD Edwards EnterpriseOne stores in configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, and so on). Beginning with the JD Edwards EnterpriseOne Tools Release 8.97, it is highly recommended that you only access and manage these settings for the supported server types using the Server Manager program. See the Server Manager Guide on My Oracle Support.

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Indicates field values.
Italics	Indicates emphasis and JD Edwards EnterpriseOne or other book-length publication titles.
`Monospace`	Indicates a JD Edwards EnterpriseOne program, other code example, or URL.

1

Introduction to JD Edwards EnterpriseOne Data Access Tools

This chapter contains the following topics:

- Section 1.1, "JD Edwards EnterpriseOne Data Access Tools Overview"
- Section 1.2, "JD Edwards EnterpriseOne Data Access Tools Implementation Steps"

1.1 JD Edwards EnterpriseOne Data Access Tools Overview

Oracle's JD Edwards EnterpriseOne Data Access Tools addresses four tools from the Oracle's JD Edwards EnterpriseOne toolset:

- Oracle's JD Edwards EnterpriseOne Table Design Aid is used to create and modify tables
- Oracle's JD Edwards EnterpriseOne Cross Reference Facility is used to determine where objects are used in the system
- Oracle's JD Edwards EnterpriseOne Business View Design Aid is used to create and modify business views
- Oracle's JD Edwards EnterpriseOne Table Conversion is used to convert tables and copy data between tables

1.2 JD Edwards EnterpriseOne Data Access Tools Implementation Steps

In the planning phase of the implementation, take advantage of all JD Edwards EnterpriseOne sources of information, including the installation guides and troubleshooting information.

The following implementation steps need to be performed before working with JD Edwards EnterpriseOne Data Access Tools:

1. Set up permissions to access and use Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW) using Oracle's JD Edwards EnterpriseOne Security Workbench.

   See "Managing Application Security" in the *JD Edwards EnterpriseOne Tools Security Administration Guide*.

2. Add yourself to the system in a developer role so that you have permissions to create Oracle's JD Edwards EnterpriseOne objects.

   See "Setting Up User Roles" in the *JD Edwards EnterpriseOne Tools Object Management Workbench Guide*.

3. Set up permissions to create OMW projects.

   See "Setting Up Allowed User Actions" in the *JD Edwards EnterpriseOne Tools Object Management Workbench Guide*.

4. Set up the appropriate database permissions so that you can add tables to the database, drop tables, and add and modify data.

   Work with the database administrator to set up these permissions.

Part I

Designing Tables in JD Edwards EnterpriseOne

2 Designing Tables

This chapter contains the following topics:

2.1 Understanding JD Edwards EnterpriseOne Table Design Aid

The Oracle's JD Edwards EnterpriseOne system uses a relational database. Tables are related using common key fields.

Database tables store the data that is used by applications in columns and rows. Each column is a data item, and each row is a record. You can create custom tables for use in JD Edwards EnterpriseOne applications.

You create tables by selecting data items from the data dictionary and assigning key fields as indices. An index enables a database management system to sort and locate records quickly. You must define a table so that the JD Edwards EnterpriseOne software recognizes that the table exists.

Use JD Edwards EnterpriseOne Table Design Aid to generate the table whenever you:

- Create new tables.
- Add or delete data items.
- Add or modify an index.

2.2 Understanding Table Creation

When an existing or custom application requires that you create tables for the JD Edwards EnterpriseOne database, use JD Edwards EnterpriseOne Table Design Aid to create the custom tables. You can access JD Edwards EnterpriseOne Table Design Aid from the Oracle's JD Edwards EnterpriseOne Object Management Workbench.

To create tables:

1. Create a table object.

Use the recommended naming conventions when creating table objects.

2. Design the table.

   Add data dictionary items as table columns and create indices. Include audit fields in the table.

3. Generate the table.

4. Add data to the table.

2.3 Creating Tables

This section discusses how to:

- Create audit trail information.
- Name tables.
- Use index guidelines.

2.3.1 Creating Audit Trail Information

Before you add a new table to the JD Edwards EnterpriseOne system, determine whether an existing table contains the data items required. If an appropriate table does not exist, you must create a new table.

When you add a new table, you should include these audit trail columns:

- User ID (USER)
- Program ID (PID)
- Machine Key (MKEY)
- Date - Updated (UPMJ)
- Time - Last Updated (UPMT)

2.3.2 Naming Tables

Use these naming conventions when adding tables:

The name of a table can be a maximum of eight characters and should be formatted as Fxxxxyyy, where:

F = Data table.

xx (second and third digits) = the system code, such as:

- 00 - Foundation
- 01 - Address Book
- 03 - Accounts Receivable

xx (fourth and fifth digits) = the group type, such as:

- 01 - Master
- 02 - Balance
- 1X - Transaction

yyy (sixth through eighth digits) = object version, such as programs that perform similar functions but vary distinctly in specific processing.

LA through LZ = Logical file.

JA through JZ = Table join.

Columns must include a two-character prefix that is used to uniquely identify the table columns. The first character must be alphabetic, while the second character can be alphanumeric. You cannot assign special characters to table columns: for example, $, #, or @. The data item alias follows the two-character column prefix. Typically, the column prefix indicates the type of data included in the table; for example, the prefix of the columns in the Address Book Master (F0101) table begin with *AB.*

A table description can be no more than 60 characters. Ensure that the table description is the topic of the table. If the table description comes from the iSeries, it should be the same name as the file that it represents, such as F0101 (Address Book Master) and F4101 (Item Master).

2.3.3 Using Index Guidelines

If an index includes only one field, use the field name as the index name: for example, Address Number.

If an index includes two fields, list them consecutively: for example, Address Number, Line Number ID.

If an index includes more than two fields, and the first two fields are the same as the first two fields of another index, list the first two fields and follow them by an alpha character: for example, Address Number, Line Number, A. Place a comma and space between each index field and between the last index field and the alpha character. Do not include more than 10 fields in an index.

The total length of the index name cannot exceed 19 characters. If you exceed 19 characters, the compiler displays a warning. This situation affects fetches that use the wrong index ID in business functions.

2.4 Working with JD Edwards EnterpriseOne Table Design Aid

This section discusses how to:

- Use Table Design Aid forms.
- Select data items.
- Define indices.
- Review table information.

2.4.1 Using Table Design Aid Forms

JD Edwards EnterpriseOne Table Design Aid presents these forms within a single window:

- Columns

  Displays the data items that you select to include in the table.
- Data Dictionary Browser

  Provides a search method to locate the data items required for the table.
- Indices

  Provides a method to define unique data items for quick sorting and updating of the table.

- Object Properties

  Displays data item attributes for a data item that is selected in the Columns form.

2.4.2 Selecting Data Items

Use data items to create table columns. Table columns store information used by applications. Data items must exist in the data dictionary before you can use them in a table. In JD Edwards EnterpriseOne Table Design Aid, you can locate data items using the query by example (QBE) line of the Data Dictionary Browser. To select data items, you can double-click them or drag them to the Columns form. All selected data items appear on the Columns form. Tables can contain data items from multiple system codes.

When you modify or delete data items, you must regenerate the table. Changes that you make to a table can affect business views and forms that reference that table.

Use the Generate Table feature to generate a newly modified table. Generating an existing table clears all data from the table.

Important: If you delete a table or delete columns from a table, then business views that reference that table, or the deleted table columns, are invalid. The system displays error messages when you generate the application.

2.4.3 Defining Indices

Indices are used to locate specific records and to sort records faster. Table indices are like tabs in a card file. Each index is made up of one or more keys, which are individual data items. Use indices to access data in a simple manner, rather than to read the data sequentially.

Tables can have multiple indices, but every table must have only one primary index. The primary index is the one unique identifier for each record in the table. Additionally, you can use the primary index to build business views. The system does not allow you to save a table without defining a primary index.

When you modify or delete indices, you must regenerate the table. Changes that you make to a table can affect business views and forms that reference that table.

2.4.4 Reviewing Table Information

You can use JD Edwards EnterpriseOne Table Design Aid to review information regarding a table. You can review the table information online or send it to a printer. This table describes the table information:

Table Information	Description
Description	The description of each data item included in the table.
Type	The field type of each data item included in the table.
Len	The field length of each data item included in the table.
Alias	The alias of each data item included in the table.

Table Information	Description
Data Item	The name of each data item included in the table.
Table Column Prefix	The prefix of the table columns.
Indices	Information regarding all of the indices included in the table.

2.5 Working with Tables

This section discusses how to:

- Generate tables
- Generate indices
- Generate header files
- Copy tables
- Remove tables

2.5.1 Generating Tables

You must generate tables to create the physical table in the database. Tables are created in the database based on the specifications that you defined. After the table is generated, you can add data to the table. Table generation also creates an .h file, or header file, that is used in business functions and table event rules.

JD Edwards EnterpriseOne Object Management Workbench provides a central location from which you can manage tables.

After you have selected data items for a table and assigned indices, you can configure the table for a specific data source. JD Edwards EnterpriseOne Object Management Workbench employs the Oracle's JD Edwards EnterpriseOne Object Configuration Manager (P986110) application to configure tables. You can configure the table within any existing data source. If you do not indicate a data source, the software automatically configures the table according to the default data source mapping. You can change the path code to generate tables in a different location. Doing so causes the system to perform a drop statement, similar to the remove table, after which the system recreates the table.

After you modify a table, you must regenerate it. If you regenerate an existing table, the table data is lost. To ensure that data is not lost, you must export the data, generate the table, and then copy the data back into the generated table.

2.5.2 Generating Indices

You must regenerate indices each time that you modify existing indices or create additional indices. Doing so modifies the .h file. When you regenerate indices, you do not lose existing data as you do when you regenerate the entire table.

2.5.3 Generating Header Files

Header files, or .h files, are used in business functions and table event rules. Occasionally, you might run across a table that does not include a header file. You can generate header files without having to generate the entire table.

Header files are located in the Include folder in the path code in which the file was generated (for example, E812\DV812\Include).

2.5.4 Copying Tables

Use the Copy Table feature to copy tables from one data source to another. However, doing so does not copy the table specifications. You can also use Table Conversion to copy tables from one data source to another.

2.5.5 Removing Tables

You cannot physically delete a table using JD Edwards EnterpriseOne Table Design Aid. If you delete a table from JD Edwards EnterpriseOne Table Design Aid, the system deletes only the specifications; it does not delete the physical table.

To completely remove a table from the system, you must use the Remove Table From Database feature.

2.6 Viewing the Data in Tables

You can view data in tables using:

- Oracle's JD Edwards EnterpriseOne Data Browser on the web client.
- Oracle's JD Edwards EnterpriseOne Universal Table Browser (UTB) on the Microsoft Windows client.

Both tools enable you to verify the existence of data in tables, as well as determine the structure of the table. You can use both tools to view data in all JD Edwards EnterpriseOne supported databases, independent of the type of database that you use.

UTB is an executable application that is part of the JD Edwards EnterpriseOne Microsoft Windows client install. You cannot use JD Edwards EnterpriseOne security to control user permissions for UTB. However, you can apply form security to the Table and Data Source Selection form (W98TAMC). This action secures UTB because the executable cannot function without this form. All column and row security that you set up through the Oracle's JD Edwards EnterpriseOne application applies to UTB.

JD Edwards EnterpriseOne Data Browser is part of the Oracle's JD Edwards EnterpriseOne web client product. You can set up security for the JD Edwards EnterpriseOne Data Browser using JD Edwards EnterpriseOne Security Workbench.

See Also:

- "Managing Data Browser Security" in the *JD Edwards EnterpriseOne Tools Security Administration Guide*
- "Using the Universal Table Browser" in the *JD Edwards EnterpriseOne Tools System Administration Guide*
- "Viewing the Data in Tables and Business Views" in the *JD Edwards EnterpriseOne Tools Foundation Guide*

2.7 Designing Custom Tables

This section discusses how to:

- Add table objects.
- Modify table objects.

- Select data items for tables.
- Define indices.
- Preview tables.
- Generate tables.
- Generate indices.
- Generate header files.
- Copy tables.
- Remove tables.
- View the data in tables.

2.7.1 Forms Used to Create Tables

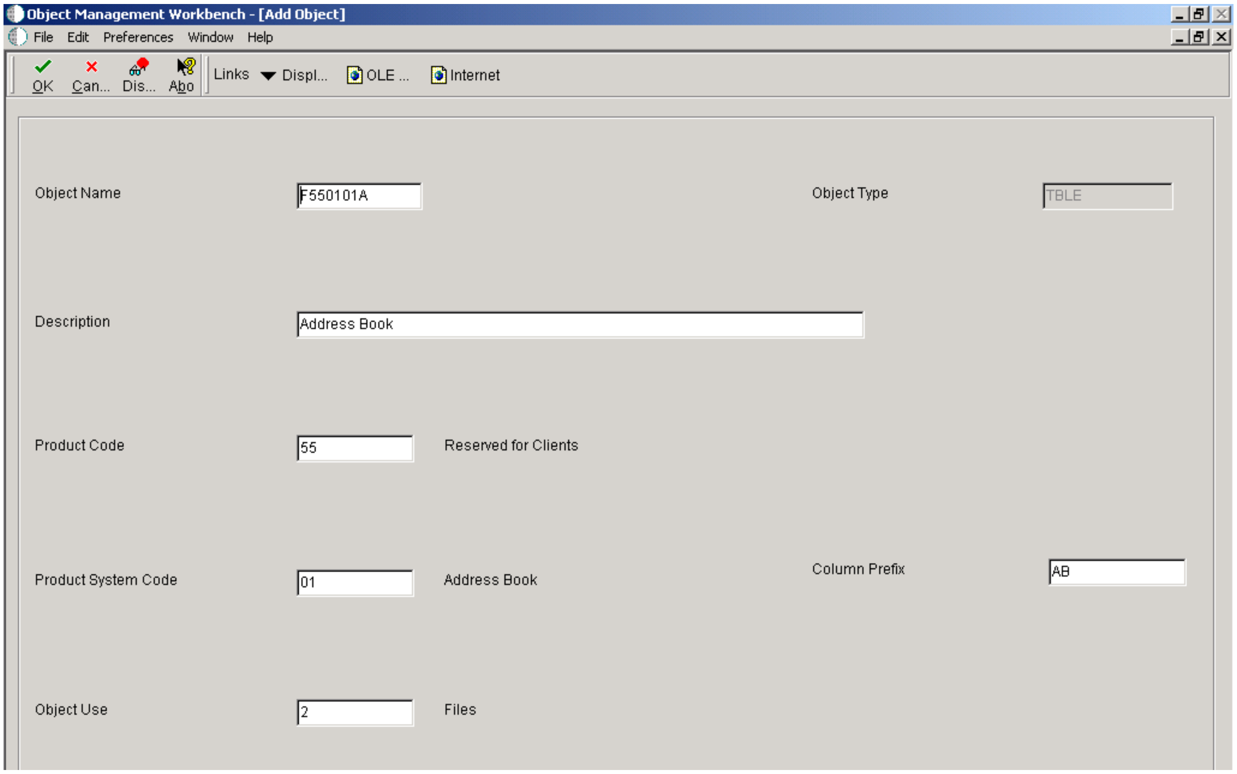
Form Name	FormID	Navigation	Usage
Object Management Workbench	W98220A	EnterpriseOne Life Cycle Tools, Application Development (GH902), Object Management, Object Management Workbench	Select and view objects in projects and access the JD Edwards EnterpriseOne design tools.
Add EnterpriseOne Object to the Project	W98220C	Click the Objects node of a project and click Add on the Object Management Workbench form.	Add new objects to a project.
Add Object	W9861AF	Select Table and click OK on the Add EnterpriseOne Object to the Project form.	Enter the table name, description, product code, product system code, object use, and column prefix.
Table Design Aid	W9860AL	Complete the object information and click OK on the Add Object form.	Access Table Design Aid, generate header files, generate tables, generate indexes, remove tables from the database, and copy tables.
Generate Table	W9866E	Select the Table Operations tab, and click Generate Table on the Table Design form.	Generate tables.
Generate Indexes	W9866J	Select the Table Operations tab, and click Generate Indexes on the Table Design form.	Generate indices.
Copy Table	W9866M	Select the Table Operations tab, and click Copy Table on the Table Design form.	Copy tables.

Form Name	FormID	Navigation	Usage
Remove Table	W9866D	Select the Table Operations tab, and click Remove Table from Database on the Table Design form.	Remove tables.
Universal Table Browser	NA	EnterpriseOne Life Cycle Tools, Application Development Tools (GH902), Object Management	View the data in tables.

2.7.2 Adding Table Objects

Access the Add Object form.

Figure 2–1 Add Object form



Object Name	F550101A		Object Type	TBLE
Description	Address Book			
Product Code	55	Reserved for Clients		
Product System Code	01	Address Book	Column Prefix	AB
Object Use	2	Files		

Object Name
Enter a unique name for the new table. Use the recommended naming convention for naming JD Edwards EnterpriseOne objects.

Description
Enter a meaningful description of the table.

Product Code
Select a user-defined code (UDC) (98/SY). Use one of the values from the range of values reserved for clients: 55–59.

Product System Code
Select a UDC (98/SY) that represents the JD Edwards EnterpriseOne system where the data is used. This value is used for reporting and jargon purposes. Example values include:

01: Oracle's JD Edwards EnterpriseOne Address Book

03B: Oracle's JD Edwards EnterpriseOne Accounts Receivable

04: Oracle's JD Edwards EnterpriseOne Accounts Payable

09: Oracle's JD Edwards EnterpriseOne General Accounting

11: Oracle's JD Edwards EnterpriseOne Multicurrency

Object Use
Select a UDC that indicates the use of the object. For example, the object may be used to create a program, a master file, or a transaction journal. A 2, or any other value from the 200 series, represents a table.

Object Type
The system displays an abbreviation that identifies the type of object being created. The object type for a table is TBLE. This field is populated by the system based on the type of object that you selected on the Add Object form.

Column Prefix
Enter a two-character prefix to be used for the table column names.

2.7.3 Modifying Table Objects

Either click OK on the Add Object form or select a table in a project and click Design to access the Table Design form.

1. Select the Summary tab and revise these fields as appropriate:
   - Description
   - Product Code
   - Product System Code
   - Object Use
2. Select the Attachments tab and enter information in the text area to document the table.

2.7.4 Selecting Data Items for Tables

Select a table in a project and access JD Edwards EnterpriseOne Table Design Aid.

1. On the Data Dictionary Browser form, use the query by example (QBE) line to locate the required data dictionary items to include in the table.
2. Drag each required data dictionary item from the Data Dictionary Browser to the Columns form.
3. To remove a column from a table, select the column and select Delete from the Edit menu.

2.7.5 Defining Indices

Select a table in a project and access JD Edwards EnterpriseOne Table Design Aid.

1. Click the indices form so that it is active, and the indices menu is visible.
2. From the indices menu, select Add New.

   You can also drag indices from the Columns form onto the Indices form.

   The index description is Untitled, and the index is marked with a key icon that displays the letter *P* to indicate a primary index.
3. Enter a name for the index, and press Enter.
4. On the Columns form, drag appropriate columns to the index.

   A unique index is marked with a single key. You can right-click the index and select Unique from the Index menu to toggle the unique status. The Unique Primary Index cannot be changed to a non-unique status.
5. Right-click the data item and select or clear the Ascending option to indicate the sort order of ascending or descending for an index column.

   An upward-pointing arrow indicates that the index column is sorted in ascending order.

2.7.6 Previewing Tables

Select a table in a project and access JD Edwards EnterpriseOne Table Design Aid.

1. Click the Columns form so that it is active, and select Print Preview from the File menu.

   A preview of the table appears on the Columns form in place of the column names.
2. On the Columns form, click Zoom In to enlarge the preview.
3. Click Print to send the information to the printer.

2.7.7 Generating Tables

Access the Generate Table form.

Figure 2–2 Generate Table form

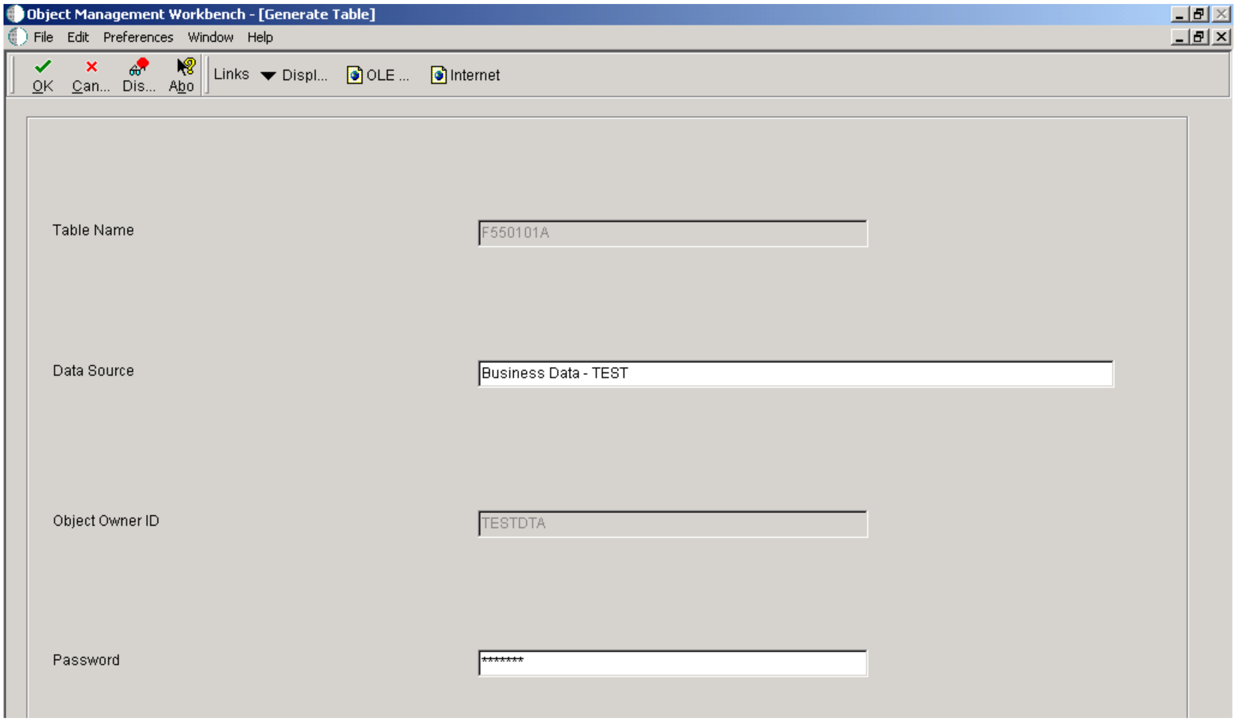


Table Name	F550101A
Data Source	Business Data - TEST
Object Owner ID	TESTDTA
Password	*******

Table Name
Displays the unique name of the table. This field is populated by the system based on the name that you gave the table when you created it.

Data Source
Enter the name of the data source where the database resides.

Object Owner ID
Enter the owner ID of the database that resides in the defined data source.

Password
Enter the database password that corresponds to the owner ID.

2.7.8 Generating Indices

Access the Generate Indexes form.

Table Name
Displays the unique name of the table. This field is populated by the system based on the name that you gave the table when you created it.

Data Source
Enter the name of the data source where the database resides.

Object Owner ID
Enter the owner ID of the database that resides in the defined data source.

Password
Enter the database password that corresponds to the owner ID.

2.7.9 Generating Header Files

Access the JD Edwards Table Design Aid form.

1. Select the Design Tools tab, and click Generate Header File.
2. The system generates a .h, or header, file.

2.7.10 Copying Tables

Access the Copy Table form.

Figure 2–3 Copy Table form

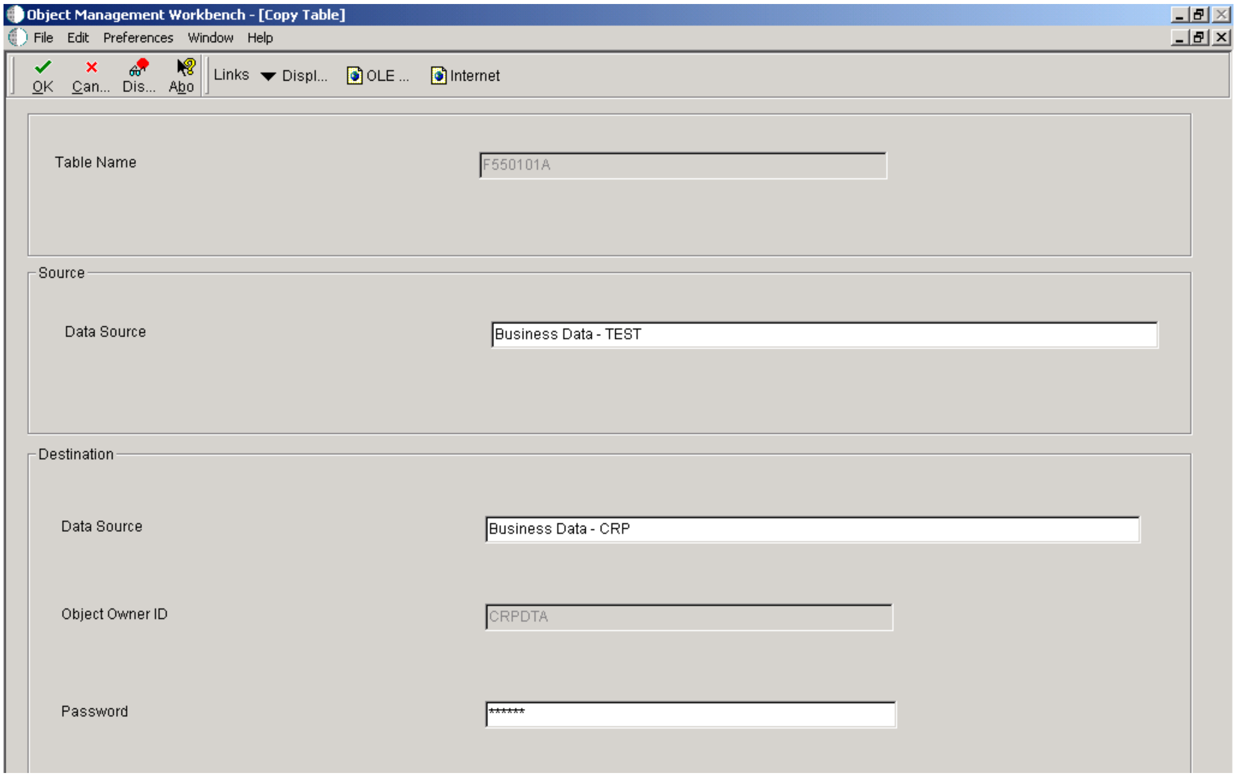


Table Name: F550101A

Source

Data Source: Business Data - TEST

Destination

Data Source: Business Data - CRP

Object Owner ID: CRPDTA

Password: ******

Table Name
Displays the unique name of the table. This field is populated by the system based on the name that you gave the table when you created it.

Source Data Source
Enter the name of the source data source, that is the data source to copy from.

Destination Data Source
Enter the name of the target data source, that is the data source to copy to.

Object Owner ID
Enter the owner ID of the database that resides in the defined data source.

Password
Enter the database password that corresponds to the owner ID.

2.7.11 Removing Tables

Access the Remove Table form.

Table Name
Displays the unique name of the table. This field is populated by the system based on the name that you gave the table when you created it.

Data Source
Enter the name of the data source where the database resides.

Object Owner ID
Enter the owner ID of the database that resides in the defined data source.

Password
Enter the database password that corresponds to the owner ID.

2.7.12 Viewing the Data in Tables

Access the Universal Table Browser form.

Table
Enter the name that identifies a table in JD Edwards EnterpriseOne. For example, F0101 is the name of the Address Book master table. You can use the visual assist to search for a table

Data Source
Enter the name of the data source where the table resides. The default value is obtained from the JD Edwards EnterpriseOne Object Configuration Manager (OCM) settings for the environment in which the user is signed on. Use the visual assist to select from a list of valid JD Edwards EnterpriseOne data sources.

Format Data
Select this option to indicate whether UTB should format the data once it is fetched from the database. JD Edwards EnterpriseOne Data Browser on the web does not provide this option. It always formats data after it is fetched from the database. The options are:

Formatted: UTB formats data according to the specifications of the JD Edwards EnterpriseOne data dictionary item. For example, assume that the data item PROC is a numeric field with a size of 15 and includes four display decimals. For a PROC value of 56.2185, UTB displays a formatted value (using the data dictionary editing) as **56.2185,** even though the value is stored in the database as 562185.

Unformatted: UTB displays the data according to the specification of the database and the data item type (such as numeric). For example, assume that the table data item PROC is a numeric field stored in the database. Depending on the type of database, the default for this field might be 32 with a precision of 15. Because JD Edwards EnterpriseOne does not store the decimals in the database, a PROC value of 56.2185 would be stored by the database as **562185.000000000000000** and would appear the same in UTB.

3 Working with Table Input/Output

This chapter contains the following topics:

3.1 Understanding Table I/O

Use the Table I/O feature in Event Rules Design to create instructions that perform table input and output. The Table I/O feature enables you to access a table through event rules. You can use the Table I/O feature to:

- Retrieve records.
- Update or delete records.
- Add records.

Table I/O event rules enable event rule support for database access.

3.1.1 Available Operations

This table describes the Table I/O features and the operations that you can perform:

Operation	Description
FetchSingle	Combines the Select and Fetch options in a basic operation. Indexed columns are used for the Select, and non-indexed columns are used for the Fetch. The operation opens a table for input or output but does not close it. All tables that do not use handles close automatically when the form that uses them is closed.
Insert	Inserts a new row.
Update	Updates an existing row. Only those columns mapped (presently in tables with or without handles) are updated. You can do partial key updates with tables and handles to tables. If you do not specify all the keys, then several records might be updated.

Operation	Description
Delete	Deletes one or more rows in a table.
Open	Opens a table or business view.
Close	Closes a table or business view.
Select	Selects one or more rows for a subsequent FetchNext operation.
SelectAll	Selects all rows for a subsequent FetchNext operation.
FetchNext	Fetches rows that you specify. You can fetch multiple records with multiple FetchNext operations or with a FetchNext operation in a loop.

3.1.2 Valid Mapping Operators

The Mapping form displays available objects that you can map to selected table columns. For SELECT statements, the available objects are used to build a WHERE clause. For FETCH statements, the available objects are used to receive data fetched from the database.

Key columns have an asterisk (*) next to them.

This table describes the operators that you can use for mapping specific table I/O operations:

Table I/O Operation	Mapping Operators
FetchSingle	Index Fields: =, <, <=, >, >=, !=, Like Non-Index Fields: Copy Target
Insert	All Fields: Copy Source
Update	Index Fields: = Non-Index Fields: Copy Source
Delete	All Fields: =
Open	NA
Close	NA
Select	All Fields: =, <, <=, >, >=, !=, Like
SelectAll	NA

3.2 Understanding Buffered Inserts

You can use buffered inserts to improve the performance of inserting hundreds or thousands of records into a single database table as long as you don't need immediate feedback if an insertion failure occurs. You cannot use buffered inserts with interactive applications but you can use them with:

- Table conversions
- Table I/O
- Batch processes
- Business functions

Buffered inserts are not available with an Access database, post-insert triggers, or multiple-table views. They are available only with these databases:

- Oracle (V8 and later)
- DB2/400
- SQL Server

The JD Edwards EnterpriseOne database middleware delivers records to the database management system one buffer load at a time.

When you request buffering, the database records are inserted individually and the buffer is automatically flushed when it fills; that is, the JD Edwards EnterpriseOne database middleware delivers the buffer to the database management system. The buffer can also be explicitly flushed. For example, the buffer flushes automatically when you commit a transaction or when you close a table or business view. The business function, table conversion engine, or Table I/O feature can explicitly request that the buffer be flushed as well.

3.2.1 Buffered Insert Error Messaging

Use caution when deciding to use buffered inserts. The system provides no immediate feedback while using buffered inserts if an insertion fails. If an insertion fails, the error appears in the log file. Consequently, buffered inserts are used primarily with batch applications.

Unless you are using the Oracle's JD Edwards EnterpriseOne Table Conversion application, you must request more detailed information from the middleware to get detailed error messages. In the JD Edwards EnterpriseOne Table Conversion application, the table conversion engine automatically performs this task. You can enable tracing to receive more detailed error messages. Otherwise, you get an error message that the insert failed. You should clear the output tables so that you do not receive duplicate error logging.

After you set up buffered inserts, you can use Special Operations to flush the buffers or get error messages.

3.3 Understanding Special Operations

In addition to the basic and advanced operations available on the Insert TableIO Operation form, there are two special operations:

- Flush Insert Buffer

  To maintain data integrity, you should flush the insert buffer before you perform any operations other than an insert. If you fail to do so, the results of recent inserts might not be reflected in other operations, and the operations might not work properly.

  When you use the flush insert buffer option for a specific table, you must flush the buffer before you close the table.

- Get Error Data

  To retrieve errors for records that the system did not insert properly, use the Get Error Data option. Depending on when the buffers are flushed, or when you begin another insert, you might overwrite the error information for a specific insert. If error information is critical, retrieve the information before the next insert begins.

If you need to perform special error handling, set the error handling up after each table I/O insert and each Flush Insert Buffer option. Always retrieve the error information *before* you begin the next table operation.

> **Note:** The return code of each table I/O statement is stored in system variable SV File_IO_Status. If an Insert or Flush Insert Buffer operation failed, the SV File_IO_Status system variable might contain a value of CO ERROR_DETAILS_AVAILABLE. If so, you need to run the Get Error Data table I/O special operation to retrieve the error data. The Get Error Data operation returns the values used in the insert for all the requested columns.

3.3.1 Get Error Data Event Rules Example

This example illustrates how you can use CO ERROR_DETAILS_AVAILABLE:

```
F0101.Insert //Attempt an insert
  SL AgingDaysAP1 -> TK Address Number
  RC Page - -⇒
⇒
⇒
⇒
⇒
⇒
⇒
> TK Tax ID
  SL TargetEnvironment -> TK Description - Compressed
If SV File_IO_⇒
Status is equal to CO ERROR_DETAILS_AVAILABLE
 F0101.Get Error Data //Failed with⇒
 errors so get errors
      //Map values used in insert to the
      //specified⇒
 fields.
   SL AgingDaysAP1 <- TK Address Number
   RC TESTT <- TK Tax ID
   SL⇒
 ReportName <- TK Description - Compressed
End If
```

3.4 Understanding Handles

In JD Edwards EnterpriseOne table I/O operations, the term *handle* refers to a type of file pointer. This file pointer connects the application or Universal Batch Engine (UBE) with the JD Edwards EnterpriseOne database middleware that communicates with the database manager. Handles point to a database table, and they are references to an address within the database middleware. Handles enable you to perform these operations, which cannot be performed using non-handle table I/O operations:

- Concurrently open multiple instances of a single table or business view.
- Open a table or business view in an environment other than the environment to which you are signed in.

  This feature is particularly helpful when you receive an upgrade to JD Edwards EnterpriseOne software or when you need to convert data from another system into JD Edwards EnterpriseOne software.

- Pass handles into a form, named event rule, or business function so that you do not need to open a table or business view more than once.

Note: You cannot use handles in transaction processing.

If you pass a handle to a form or a named event rule, a HANDLE data item must be included as a member in the data structure for the form or named event rule. In the form interconnect or business function call, you must assign a handle value from the event rules to the HANDLE data structure member. You can use this handle in the form or named event rule in the same way that you use any other table I/O handle.

You must explicitly open and close handles, unlike other table I/O operations in which the system implicitly opens or closes the table for you. You must open a handle before you can use it. All of the table I/O operations, except Open, work the same for handles as they do for tables and business views. When you are finished using a handle, you must explicitly close it. Close the handle in the same way that you close a table or business view, except select the Handle option.

In order to use handles in table I/O event rules, you must:

- Define the handle in the data dictionary.
- Create a handle variable in event rules.
- Open the handle explicitly.

You might be able to find an existing handle for the table I/O. Whether you use an existing handle or create a new handle, you must create a handle variable in Event Rules Design. Create a handle variable in the same way that you create other variables. You can use any scope that is necessary.

After you create a handle variable, you must explicitly open the handle. Then, after performing the required table I/O, you must explicitly close it.

See Also:

- "Creating a Data Dictionary Item" in the *JD Edwards EnterpriseOne Tools Development Tools: Data Dictionary Guide*

3.5 Working with Table I/O

This section lists the prerequisites and discusses how to:

- Create table I/O event rules.
- Use buffered inserts in table I/O.
- Use special operations for buffered inserts.
- Create handles.
- Use handles with table I/O.

3.5.1 Prerequisites

Before creating table I/O event rules, ensure that you:

- Create a batch application.
- If you want to update the database, ensure that the batch application is defined as an update report.

3.5.2 Creating Table I/O Event Rules

Open a batch application in Oracle's J.D. Edwards EnterpriseOne Report Design Aid (RDA) and access Event Rules Design.

1. Click the Table I/O button.
2. On the Insert TableIO Operation form, select an operation from Basic Operations, Advanced Operations, or Special Operations and click Next.
3. On the Data Source form, select a table, business view, or handle as the data source and click Next.
4. On the Mapping form, select fields from the Available Objects list to map to columns in the mapping.

   The key field must be mapped. Map as many other fields as required to ensure that the correct data is associated with the correct record.
5. Click the Operation for each map until you locate the operator that you want.

   The default operator is equal.
6. Click Finish to save the operation and return to Event Rules Design.

3.5.3 Using Buffered Inserts in Table I/O

Open a batch application in RDA and access Event Rules Design.

1. Click the Table I/O button.
2. On the Insert TableIO Operation form, select Open under Advanced Operations and click Next.
3. On the Data Source form, select the table that you want to use and click Advanced Options.
4. On the Advanced Options form, select Buffer Inserts and click OK.

3.5.4 Using Special Operations for Buffered Inserts

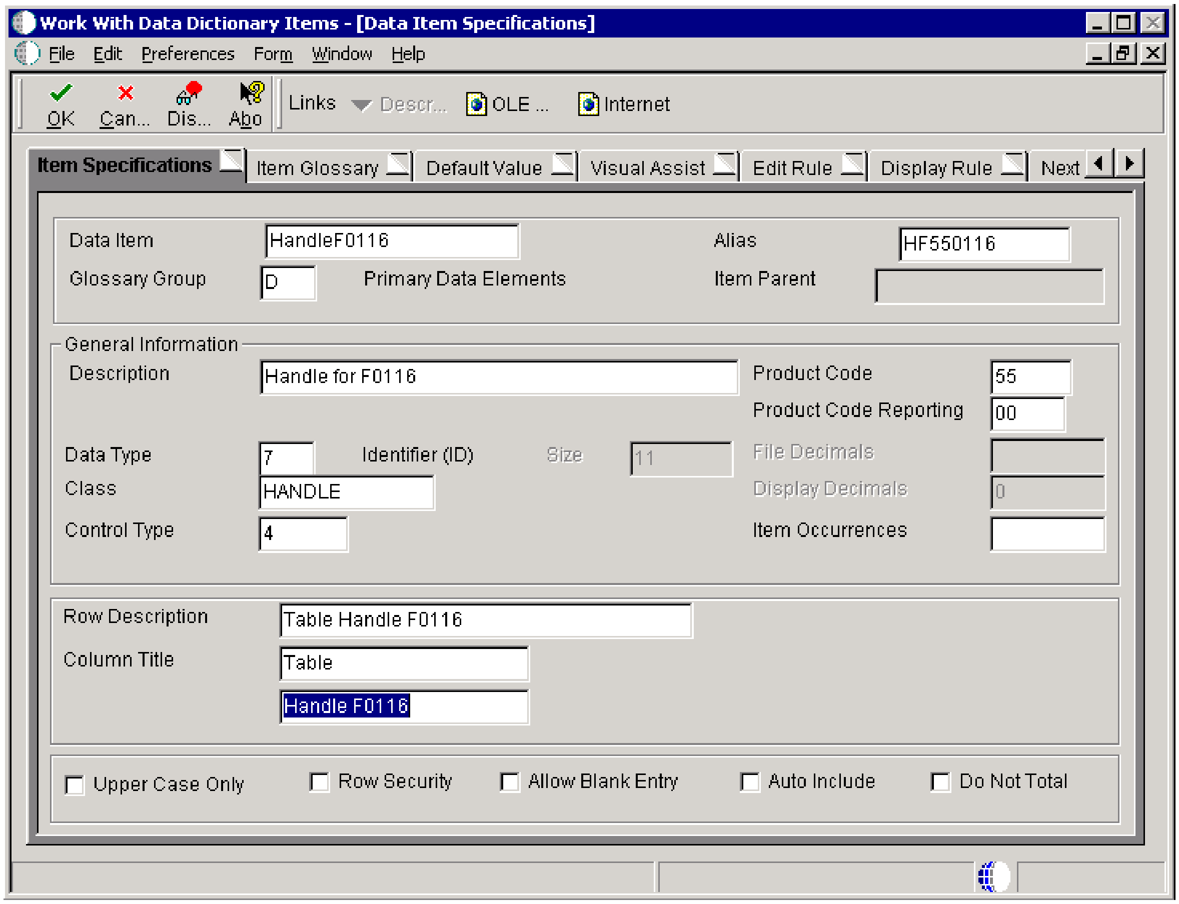
Open a batch application in RDA and access Event Rules Design.

1. At the point in which you want to perform either a Flush Insert Buffer or Get Error Data operation, click the Table I/O button.
2. On the Insert TableIO Operation form, select one of these options under Special Operations:
   – Flush Insert Buffer
   – Get Error Data

3.5.5 Creating Handles

In JD Edwards Solution Explorer access EnterpriseOne Life Cycle Tools, Application Development, Data Dictionary Design, select Work With Data Dictionary Items to access the Work With Data Items form.

Figure 3–1 Data Item Specifications form



Data Item HandleF0116

Alias HF550116

Glossary Group D Primary Data Elements

Item Parent

General Information

Description Handle for F0116

Product Code 55

Product Code Reporting 00

Data Type 7 Identifier (ID)

Size 11

File Decimals

Class HANDLE

Display Decimals 0

Control Type 4

Item Occurrences

Row Description Table Handle F0116

Column Title Table

Handle F0116

☐ Upper Case Only ☐ Row Security ☐ Allow Blank Entry ☐ Auto Include ☐ Do Not Total

Data Item

Enter the name of the data item. This is a 32-character alphabetical field that does not allow spaces or special characters.

Alias

Enter a unique identifier.

Glossary Group

Select a user-defined code (UDC) (H95/DG) that indicates the type of data item. A handle is glossary group D, Primary Data Elements.

Description

Enter a meaningful description of the data item.

Product Code

Select a UDC (98/SY) from the range of values reserved for clients: 55–59.

Product System Code

Select a UDC (98/SY) that represents the JD Edwards EnterpriseOne system where the data is used. This value is used for reporting and jargon purposes. Example values include:

01: Oracle's JD Edwards EnterpriseOne Address Book

03B: Oracle's JD Edwards EnterpriseOne Accounts Receivable

04: Oracle's JD Edwards EnterpriseOne Accounts Payable

09: Oracle's JD Edwards EnterpriseOne General Accounting

11: Oracle's JD Edwards EnterpriseOne Multicurrency

Data Type
Select the style or classification of data. Examples include, string, date, character, and identifier. A handle is an identifier data type.

Size
Enter the field size of the data item.

Class
Select a class that defines the essential attributes and characteristics of the data item. Examples include, name, email, and table handle.

Control Type
Select the type of graphical user control that is associated with the data item. Examples include button, radio button, and user-defined code.

Row Description
Enter the description of the data item as it will appear in a group report.

Column Title
Enter the description of the data item as it will appear in a columnar report.

3.5.6 Using Handles with Table I/O

Open a batch application in RDA and access Event Rules Design.

1. Create an event rule variable using the appropriate handle data dictionary item.
2. Click the Table I/O button.
3. Select Open under Advanced Operations and click Next.
4. On the Data Source form, select the Handles tab.
5. Select the handle that you want to open and click Next.
6. Select a variable that contains the name of the environment in which you want to open the table and click Finish.

   If you want to open the table in the login environment, select the system value **SL LoginEnvironment.** System values also exist for the source and target environments if you use the Table I/O in Table Conversion feature.
7. Click the Table I/O button.
8. Select Close and click Next.
9. On the Data Source form, select the Handles tab.
10. Select the handle that you want to close, and click Finish.

4 Creating Table Event Rules

This chapter contains the following topics:

- Section 4.1, "Understanding Table Event Rules"
- Section 4.2, "Working with Table Event Rules"

This chapter provides an overview of table event rules and discusses how to work with table event rules.

4.1 Understanding Table Event Rules

Use table event rules to attach database triggers (or programs) that automatically run whenever an action occurs against the table. An action against a table is called an event. When you create a database trigger in Oracle's JD Edwards EnterpriseOne, you must first determine which event activates the trigger and then use Event Rules Design to create the database trigger.

Table event rules provide embedded logic at the table level. Table event rules have their own location, events, and system functions. When you use table event rules, neither the calling application nor the user is notified of changes or events to the table. No form or report interconnect is available with table event rules.

You can use table event rules for data integrity. For example, when you delete a record in Oracle's JD Edwards EnterpriseOne Address Book, you might want to delete all associated records, such as phone and category codes. You can also use table event rules for currency. The *Currency Conversion is On* event rule handles currency information in table event rules.

You can attach event rules on a table-by-table basis to these events:

- After Record is Deleted
- After Record is Fetched
- After Record is Inserted
- After Record is Updated
- Before Record is Deleted
- Before Record is Inserted
- Before Record is Updated
- Currency Conversion is On

This table describes the functions that you can use in table event rules:

Functions	Description
Assignment\Expression	Enables you to create assignments and complex expressions.
If\While	Enables you to create IF/WHILE conditional statements.
Business Function	Enables you to call existing business functions.
System Function	Enables you to attach existing JD Edwards EnterpriseOne system functions.
Variables	Enables you to create Event Rule variables.
Else	Enables you to insert an ELSE clause, which is only valid within the boundaries of IF and ENDIF.
Table I/O	Enables you to create event rules used for database access. Performs table input and output, data validations, and record retrieval.

You do not need to create and associate data structures to table event rule functions. The table itself is the data structure that is passed to the table event rule function.

4.2 Working with Table Event Rules

This section provides an overview of table event rule triggers and discusses how to add table event rules to table conversions.

4.2.1 Understanding Table Event Rule Triggers

To create table event rules, you must:

- Create the table trigger in Event Rules Design.
- Generate the JD Edwards EnterpriseOne table trigger as C code.
- Build the table trigger.

  When you build table triggers, the system performs these actions:

  - Converts the event rule to C source code, which creates the files OBNM.c and OBNM.hxx (OBNM = Object Name).

    The source file contains one function per table event rule event.

  - Creates a make file to compile the generated code.
  - Runs the make file to compile the new functions and to add them into JDBTRIG.DLL.

    This consolidated DLL contains table event rule functions.

4.2.2 Adding Table Event Rules to Table Conversions

Access the Object Management Workbench form.

1. Check out the table to which you want to attach event rules and click Design.
2. On the Table Design form, select the Design Tools tab and click Start Table Trigger Design Aid.
3. Select an event from the Events list.

4. On Event Rules Design, click one of the event rule buttons and complete the event rules.
5. Click Save to save the event rule specifications and then click Close.
6. If you are creating a *new* table in Oracle's JD Edwards EnterpriseOne Table Design Aid, select the Table Operations tab and click Generate Table.

   Important: Never perform this step on an existing table because it clears all data.

7. On the Generate Table form, complete these fields and click OK:
   - Data Source
   - Password
8. On the Table Design form, select the Design Tools tab and click Build Table Triggers.
9. To review a log of the build, select Generate Header File, and then open the file that the system created.

   The creation of the table event rules is complete. The newly created or modified table event rule is called from the JD Edwards EnterpriseOne database middleware whenever the corresponding event occurs against the table.

Part II

Using Cross Reference Facility

5 Understanding the Cross Reference Facility

This chapter contains the following topics:

- Section 5.1, "Understanding the Cross Reference Facility"
- Section 5.2, "Searching for Objects"
- Section 5.3, "Viewing Field Relationships"
- Section 5.4, "Rebuilding Cross-Reference Information"
- Section 5.5, "Working with the Cross Reference Facility Application"

5.1 Understanding the Cross Reference Facility

Use the Oracle's JD Edwards EnterpriseOne Cross Reference Facility to determine where specific objects are used. The JD Edwards EnterpriseOne Cross Reference Facility can be accessed on the Microsoft Windows client or on the web client. However, not all features of the JD Edwards EnterpriseOne Cross Reference Facility are available on the web client.

In the JD Edwards EnterpriseOne Cross Reference Facility, you can view relationships between objects and their components. For example, you can use JD Edwards EnterpriseOne Cross Reference Facility to:

- Identify each instance in which a business function is used.
- View a list of forms within an application.
- Display all fields within a business view.
- Cross-reference all applications in which a specific field is used.

Caution: JD Edwards EnterpriseOne Cross Reference Facility is not automatically updated when you add or modify objects. You must rebuild relationships to include new information.

5.2 Searching for Objects

This section discusses how to search for JD Edwards EnterpriseOne objects.

5.2.1 Searching for Objects

You can search for these objects by object name:

- Data items

- Interactive applications
- Batch applications
- Business functions
- Business views
- Data structures
- Tables
- Forms
- Event rules

5.2.1.1 Searching for Data Items

You can determine where specific data items are used by clicking the appropriate links under these headings:

- Properties
  - Edit Rules Function Called by a Data Item.
  - Display Rule Function Called by a Data Item.
  - Search Form Used by Data Items.
  - Functions Called by a Smart Field Data Item.
- Interactive Applications
  - Forms Using a Data Item.
  - Interactive Applications Using a Data Item.
  - Interactive Applications Using a Data Item as a Variable.
- Batch Applications
  - Batch Applications Using a Data Item.
  - Batch Applications Using a Data Item as a Variable.
- Tables
  - Tables Using a Data Item.
  - Indices Using a Data Item.
  - Business Views Using a Data Item.
  - Table Event Rules Using a Data Item.
- Data Structures
  - Processing Options Using a Data Item.
  - Generic Text Data Structures Using a Data Item.
  - Business Function Data Structures Using a Data Item.
  - Add Data Structures Using a Data Item.
- Names Event Rules
  - Named Event Rules Using a Data Item.
  - Named Event Rules Using a Data Item as a Variable.

5.2.1.2 Searching for Interactive Applications

You can locate a variety of information about interactive applications by clicking the appropriate links under these headings:

- Properties
  - Tables Used by an Interactive Application.
  - Indices used by an Interactive Application.
  - Business Views Used by an Interactive Application.
  - Data Items Used by an Interactive Application.
  - Data Items Used as Variables by an Interactive Application.
  - Forms for an Interactive Application.
  - Processing Options Used by an Interactive Application.
  - Data Structures Used by an Interactive Application.
  - Generic Text Data Structures Used by an Interactive Application.
- Calling
  - Interactive Applications Calling an Interactive Application.
  - Named Event Rules Calling an Interactive Application.
- Called By
  - Interactive Applications Called by an Interactive Application.
  - Batch Applications Called by an Interactive Application.
  - Business Functions or NERs Called by an Interactive Application.

5.2.1.3 Searching for Batch Applications

You can locate information about UBE applications (UBEs) and how they are used by clicking the appropriate links under these headings:

- Properties
  - Tables Used by a Batch Application.
  - Business Views Used by an Interactive Application.
  - Indices Used by a Batch Application.
  - Data Items Used by a Batch Application.
  - Data Items Used as Variables by a Batch Application.
  - Data Structures Used by a Batch Application.
  - Processing Options Used by an Batch Application.
  - Generic Text Data Structures Used by a Batch Application.
- Calling
  - Interactive Applications Calling a Batch Application.
  - Named Event Rules Calling a Batch Application.
  - Batch Applications Calling a Batch Application.
- Called By

  - Batch Applications Called by a Batch Application.
  - Business Functions or NERs Called by a Batch Application.

5.2.1.4 Searching for Business Functions

You can locate a variety of information about business functions and how they are used by clicking the appropriate links:

- Everywhere a Business Function or NER is Used.
- Search C Source and C Header Files by String.
- NER Properties:
  - Data Structures used by a Named Event Rule.
  - Indices Used by a Named Even Rule.
  - Tables Used by a Named Event Rule.
  - Data Items Used as Variables in a Named Event Rule.
- Called by:
  - Business Functions or NERs Called by a Named Event Rule.
  - Business Functions or NERs Called by a Business Function.
  - Interactive Applications Called by a Named Event Rule.
  - Batch Applications Called by a Named Event Rule.
- Business Function Properties:

  Tables Used by a Business Function
- Calling:
  - Named Event Rules Calling a Business Function or NER.
  - Business Functions Calling a Business Function or NER.
  - Interactive Applications Calling a Business Function or NER.
  - Batch Applications Calling a Business Function or NER.
  - Table Event Rules Calling a Business Function or NER.
  - Smart Field Data Items Calling a Business Function or NER.
  - Data Items Calling an Edit Rule Business Function or NER.
  - Data Items Calling a Display Rule Business Function or NER.

5.2.1.5 Searching for Business Views

You need to click the appropriate links under the following headings to locate a variety of information about business views and understand how they are used:

- Properties
  - Data Items for a Business View.
  - Tables for a Business View.
- Use
  - Everywhere a Business View is Used.
  - Interactive Applications Using a Business View.

  - Batch Applications Using a Business View.

5.2.1.6 Searching for Data Structures

You need to click the appropriate links under the following headings to locate a variety of information about data structures and understand how they are used.

- Properties
  - Data Items Used by All Data Structures.
  - Data Items Used by a Business Function or NER Data Structure.
  - Data Items Used by a Processing Option Data Structure.
  - Data Items Used by a Generic Text Data Structure.
- Use
  - Everywhere a Data Structure is Used.
  - Interactive Applications Using a Data Structure.
  - Business Functions or NERs Using a Data Structure.
  - Batch Applications Using a Data Structure.
  - Table ER Using a Data Structure.
  - Interactive Applications Using a Generic Text Data Structure.
  - Batch Applications Using a Generic Text Data Structure.

5.2.1.7 Searching for Tables

You need to click the appropriate links under the following headings to locate a variety of information about tables and understand how they are used:

- Properties
  - Data Items Used by a Table.
  - Indices for a Table.
  - Data Items Used by an Index.
- Table Use
  - Everywhere a Table is Used.
  - Business Views using a Table.
  - Interactive Applications Using a Table.
  - Batch Applications Using a Table.
  - Named Event Rules Using a Table.
  - Forms Using a Table.
  - Table Event Rules Using a Table.
  - Business Functions Using a Table.
- Index Use
  - Interactive Applications Using an Index.
  - Forms Using an Index.
  - Batch Applications Using an Index.

  - Table Event Rules Using an Index.
  - Named Event Rules Using an Index.
- Table Event Rules
  - Tables Using Table Event Rules.
  - Data Items Used by a Table Event Rule.
  - Tables Used by a Table Event Rule.
  - Business Functions or NERs Used by Table Event Rules.
  - Data Structures Used by Table Event Rules.
  - Indices Used by a Table Event Rule.

5.2.1.8 Searching for Forms

You need to click the appropriate links under the following headings to locate a variety of information about forms and understand how they are used:

- Properties
  - Tables Used by a Form.
  - Data Items Used by a Form.
  - Indices Used by a Form.
  - Business Views Used by a Form.
- Use
  - Interactive Applications Calling a Form.
  - Data Items Using a Search Form.

5.2.1.9 Searching for Event Rules

You can search where a particular control is used in the Event Rules of an application.

The Oracle's JD Edwards EnterpriseOne BrowsER functionality found on the row menu of the ER Search form is not available on the web client. This functionality must be used on the Microsoft Windows client.

The Rebuild Relationships functionality found on the form menu of the ER Search form is not available on the web client. This functionality must be used on the Microsoft Windows client.

5.3 Viewing Field Relationships

This section discusses how to view field relationships.

5.3.1 Search Types

The Field Relationships form is meaningful only for these cross-reference search types:

Search Type	Description
DA	Indicates data items that are used by an application.
FA	Indicates forms for an application.
FI	Indicates forms that use a data item.

Search Type	Description
SA	Indicates data structures for an application.

5.3.2 Control Types

The Field Relationships form displays the control type for a field. This table describes the control types:

Control Type	Description
BC	Indicates business view columns
FI	Indicates form interconnects
GC	Indicates grid controls
FC	Indicates form controls

5.4 Rebuilding Cross-Reference Information

This section provides an overview of rebuilding cross-reference information and discusses how to rebuild information for the JD Edwards EnterpriseOne Cross Reference Facility.

5.4.1 Understanding Rebuilding Cross-Reference Information

When developers create and modify objects, the cross-reference information becomes out of date. Because the cross-reference files are not automatically rebuilt when objects are created and modified, you must periodically rebuild the cross-references manually. You can also regularly schedule cross-reference builds to ensure that the information is current. You must rebuild the cross-reference information from the Microsoft Windows client.

5.4.2 Rebuilding Information for the Cross-Reference Facility

The JD Edwards EnterpriseOne Cross Reference application contains multiple forms. Each form contain a grid of cross-reference records. Each row contains a date column (the far right column) that indicates on which date the cross-reference record was built. Use this date to verify if cross-reference records are out of date. If information is out of date, use the Rebuild option from any of the cross-reference forms on the Microsoft Widows client.

Cross-reference builds use relational database tables, not local specifications.

5.5 Working with the Cross Reference Facility Application

This section lists the pages used to work with the JD Edwards EnterpriseOne Cross Reference Facility and discusses how to:

- Search for objects
- Search for event rules
- View field relationships
- Rebuild cross-reference information

5.5.1 Forms Used to Work with the Cross Reference Facility

Form Name	FormID	Navigation	Usage
Cross Reference	W980011K	EnterpriseOne Life Cycle Tools, Application Development (GH902), Object Management, Cross Reference Facility	Search for objects and rebuild cross-reference information.
ER Search	W980011A	Select Event Rules from the Form menu on the Cross Reference form.	Search for event rules.
Field Relationships	W980011B	Select Field Relationships from the Form menu on the Cross Reference form.	View field relationships.

5.5.2 Searching for Objects

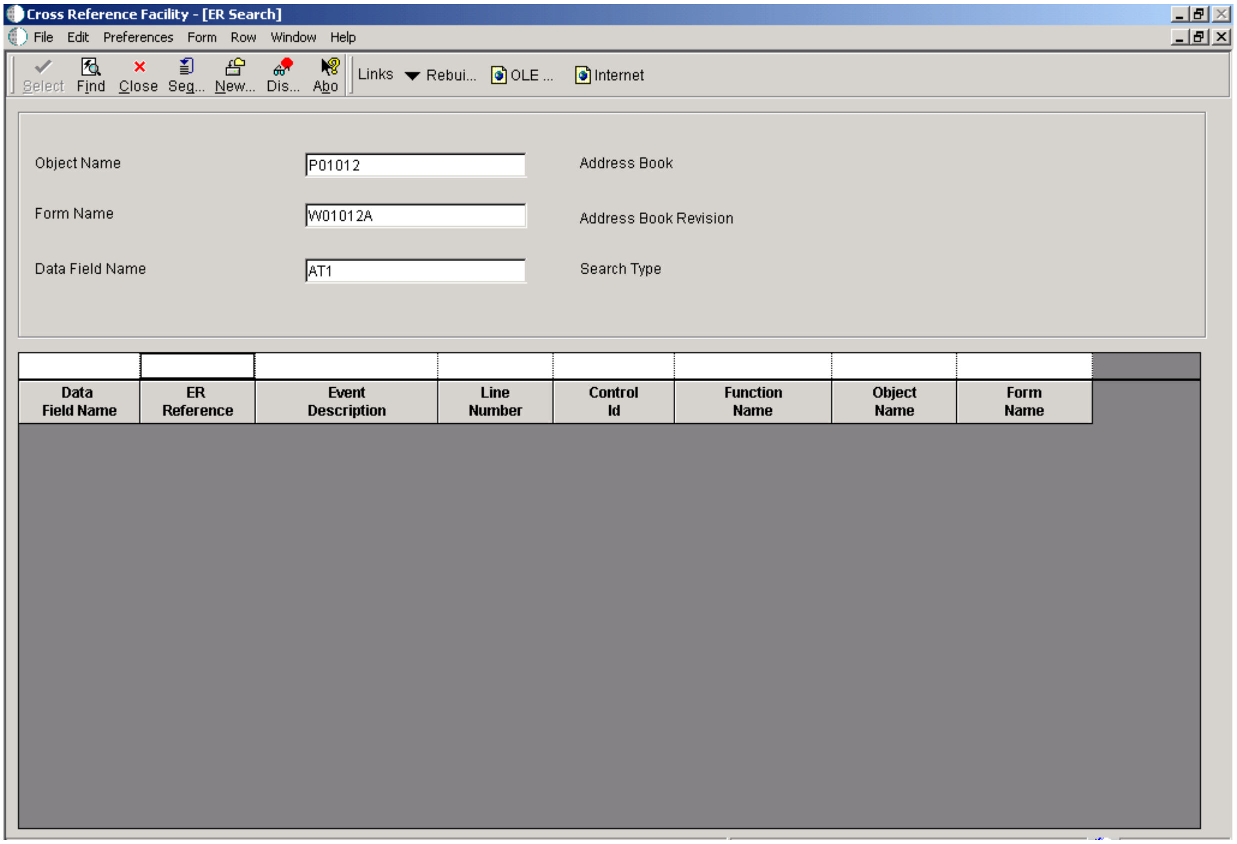
Access the Cross Reference form.

1. Select the Data Items tab, select a search option, and, from the appropriate form, enter the name of the data item on which to search.
2. Select the Interactive Applications tab, select a search option, and, from the appropriate form, enter the name of the interactive application on which to search.
3. Select the Batch Applications tab, select a search option, and, from the appropriate form, enter the name of the batch application on which to search.
4. Select the Business Functions tab, select a search option, and, from the appropriate form, enter the name of the business function on which to search.
5. Select the Business Views tab, select a search option, and, from the appropriate form, enter the name of the business view on which to search.
6. Select the Data Structures tab, select a search option, and, from the appropriate form, enter the name of the data structure on which to search.
7. Select the Tables tab, select a search option, and, from the appropriate form, enter the name of the table on which to search.
8. Select the Forms tab, select a search option, and, from the appropriate form, enter the name of the form on which to search.

5.5.3 Searching for Event Rules

Access the ER Search form.

Figure 5–1 ER Search form


Cross Reference Facility - [ER Search]
File Edit Preferences Form Row Window Help
Select Find Close Seq... New... Dis... Abo
Links Rebui... OLE ... Internet

Object Name	P01012	Address Book
Form Name	W01012A	Address Book Revision
Data Field Name	AT1	Search Type

Data Field Name	ER Reference	Event Description	Line Number	Control Id	Function Name	Object Name	Form Name

Object Name
Enter the name of the application for which you want to search for event rules. For example, if you are searching for how Search Type is used in the address book application event rules, enter **P01012** in this field.

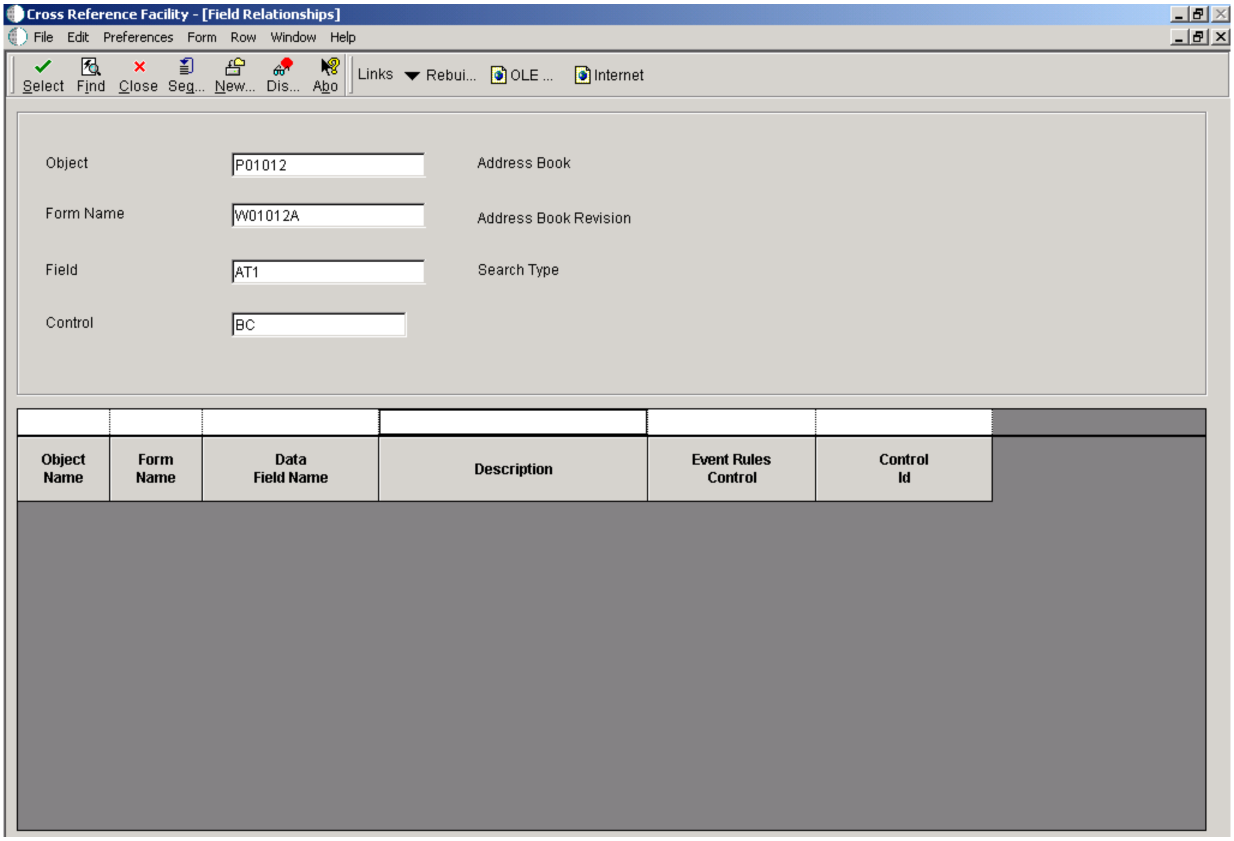
Form Name
Enter the name of the form for which you want to search for event rules. For example, if you are searching for how Search Type is used in the Address Book Revisions form, enter **W01012A** in this field.

Data Field Name
Enter the alias of the data item for which you want to search. For example, if you are searching for how **Search Type** is used in the Address Book Revisions form, enter the data dictionary item name for Search Type, which is AT1.

5.5.4 Viewing Field Relationships

Access the Field Relationships form.

Figure 5–2 Field Relationships form


Cross Reference Facility - [Field Relationships]
File Edit Preferences Form Row Window Help
Select Find Close Seq... New... Dis... Abo
Links Rebui... OLE ... Internet

Object	P01012	Address Book
Form Name	W01012A	Address Book Revision
Field	AT1	Search Type
Control	BC	

Object Name	Form Name	Data Field Name	Description	Event Rules Control	Control Id

Object
Enter the name of the application for which you want to locate field relationships.

Form Name
Enter the name of the form for which you want to locate field relationships.

Field
Enter the alias of the data dictionary item name for which you want to locate field relationships.

Control

Select the type of object for which you are searching. Values are:

- **BC:** Business View Field.
- **PO:** Processing Option.
- **FI:** Form Interconnect.
- **VAR:** Variable.
- **FC:** Form Control.
- **FCW:** Form Control Work Field.
- **GC:** Grid Column.
- **GCW:** Grid Column Work Field.

5.5.5 Rebuilding Cross-Reference Information

Access the Cross Reference form.

1. Select Rebuild Relationships from the Form menu.
2. Select the objects that you want to rebuild.

Note: The rebuild process can take several minutes.

Part III

Designing Business Views

6 Understanding Business Views

This chapter contains the following topic:

- Section 6.1, "Business Views"

6.1 Business Views

A business view is a selection of data items from one or more tables. After you create a table, use Oracle's JD Edwards EnterpriseOne Business View Design to create a business view that contains only the data items that are required for the application. Oracle's JD Edwards EnterpriseOne uses the business view that you create to generate the appropriate SQL statements that are necessary to retrieve data from the database. After you define a business view, you can attach it to a:

- Form that updates data in an interactive application.
- Report that displays data.

Business views increase performance by moving less data across the network because you include only those data items that an application requires.

Business views are required for creating applications and generating reports. Business views:

- Link a JD Edwards EnterpriseOne application to one or more tables.
- Contain all or a subset of data items from one or more tables.
- Can use table joins to join multiple tables on common fields.
- Are building blocks for text search indexes that enable full-text searching of data.

  To make a business view available for full-text searching, select the Text Search option on the properties form. Do not select this option if you do not plan to use the business view for full-text searching; doing so can negatively affect performance.

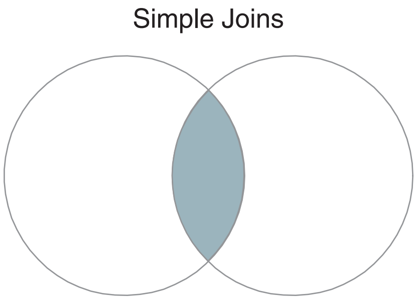
6.1.1 Table Joins

Use the table join feature to join multiple tables in a business view. Joining tables enables you to combine fields from different tables for each record of the primary table. Perform the join using fields that are common to the tables. Define the joined fields to satisfy a join condition, such as when the records, or rows, have the same value in the key fields. The primary table is the table where you initiate the join (usually the table on the left in Oracle's JD Edwards EnterpriseOne Table Design Aid) and the secondary table is the table where you conclude the join (usually the table on the right in JD Edwards EnterpriseOne Table Design Aid). Several types of joins exist, including:

Join Type	Description
Simple join, also known as inner join.	Includes only rows that match both the primary and secondary tables.
Right outer join.	Includes rows that are common to both the primary and secondary tables, and unmatched rows from the secondary table.
Left outer join.	Includes rows that are common to both the primary and secondary tables, and unmatched rows from the primary table.
SQL 92 left outer join.	Includes rows that are common to both the primary and secondary tables, unmatched rows from the primary table, and any rows with null values from the secondary table—regardless of any Where clause against the fields from the secondary table.

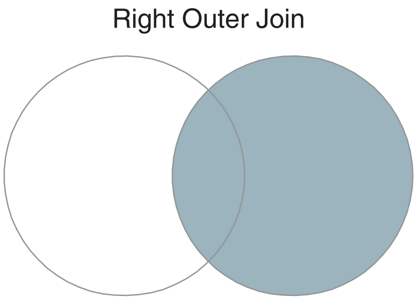
This diagram illustrates a simple table join:

Figure 6–1 Simple join



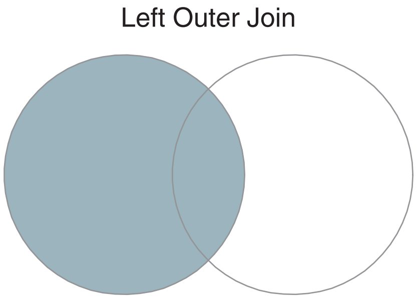
This diagram illustrates a right outer join:

Figure 6–2 Right outer join



This diagram illustrates a left outer join:

Figure 6–3 Left outer join

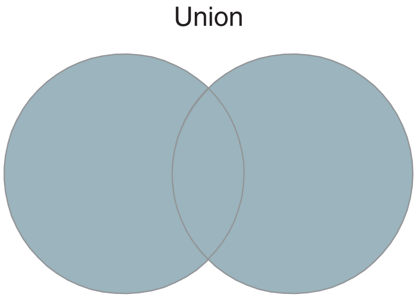


6.1.2 Table Union

A table union joins entire tables. The system first checks for rows from the primary table, and then for rows with corresponding columns from the secondary table. If the rows from the two tables contain identical data, then only one of the records is retrieved in the union. Unions include rows from the primary table and corresponding columns from the secondary table.

This diagram illustrates a union:

Figure 6–4 Table union



6.1.3 Select Distinct

If a business view includes the primary key fields of the primary table, every row of the business view query is unique. The primary key field has a different value in each row, or record, of the primary table. If the business view does not contain all primary key fields of the primary table, then duplicate rows can occur during the business view query. You can eliminate the duplicate rows in the output by using the Select Distinct feature when designing the business view.

6.1.4 Primary Key Fields

The fields that are included in the primary index of a table are displayed as key fields in JD Edwards EnterpriseOne Business View Design Aid. These fields include a key icon next to the field name. The primary key fields of a table are always included in the business view. Business views then carry information from the table to an application. To carry forward additional information other than the primary key fields, select additional fields to include in the business view.

7 Designing Business Views

This chapter contains the following topics:

7.1 Understanding Business View Design Aid

Before you begin designing a business view, consider the requirements. Determine the purpose of the application that needs the business view and the data items that are required. You can then identify the tables in which the required data items reside. Adding a new business view does not affect performance; however, using an existing business view that contains many columns that are not required by the application might negatively affect performance.

Business views usually contain a few more fields than are used on the form, in the grid, or in the batch application. The unused fields are related to the fields that are required. If requirements change, these fields can be added to the application without redesign of the business view.

You can modify business views to reflect changes in business requirements. You can easily add fields to existing business views. Deleting fields from business views, however, is more complicated. If you need to delete fields from a business view, ensure that those fields are not currently in use in an interactive or batch application.

You can use different business views for each form type that is included in an interactive application. Typically, search & select forms include the minimum number of fields that are required to keep them at a nominal size. Search & select forms should include:

- Only the fields that are necessary for filtering searches.
- The associated output fields, such as descriptions.

Find/browse and parent/child forms include more fields and are typically larger than search & select forms. However, you should still limit the number of fields on find/browse and parent/child forms to include only those fields that are required for filtering and displaying the associated output fields.

Input-capable forms are usually large and include all of the fields from the table. They should include all of the fields that are necessary to add or update a record, including audit information.

7.2 Understanding Business View Naming Conventions

When naming a business view object, guidelines indicate that the name can be a maximum of eight characters and should be formatted as:

VzzzzzzA,where:

V is the business view.

zzzzzz is the characters that represent the primary table.

A is a letter that designates the sequence of the business view.

For example, V0101A is the first business view that is created over the F0101 table, V0101B is the second business view that is created over the same table, and V0101C is the third business view that is created.

7.2.1 External Developer Considerations for Naming Business Views

The term *external development* refers to applications that are created by developers who are not employed by Oracle's JD Edwards, such as consultants who create custom applications for clients. To prevent interference between Oracle's JD Edwards EnterpriseOne and non-JD Edwards EnterpriseOne objects, you must use caution when naming an external business view. When you create business views for an application that is created by external developers, format the business view name as:

Vssss9999,where:

V is the business view.

ssss is the system code for the enterprise.

9999 is a unique next number or character pattern within the enterprise.

Provide a business view description with a maximum of 60 characters. It should reflect the application description followed by the form type, for example, Item Master Browse and Item Master Revisions.

Primary key fields should remain in the business view and should not be reorganized. If you try to remove a primary key field, you receive an error message.

Note: At least one business view for each table should include all columns from the table. Only one business view is allowed for each form type, except for a header/detail form. You can use two business views on header/detail forms, one for the header portion of the form and one for the detail portion.

7.2.2 Naming Joined Business Views

The description of a joined table business view should include the names of the tables that are being joined. Place the primary table first, and separate the table names using a forward slash (/).

For example, create a joined table business view by joining the F4101 (Item Master) table and the F4102 (Item Branch) table. The F4101 table is the primary table, so the business view should be named:

Item Master and Branch F4101/F4102.

7.3 Understanding Business View Design Aid

This section provides an overview of the JD Edwards EnterpriseOne Business View Design Aid interface and discusses:

- Tables for business views.
- Data items for business views.
- Select Distinct.
- Table joins.
- Table unions.

7.3.1 Understanding the Business View Design Aid Interface

This table describes the forms presented in the single JD Edwards EnterpriseOne Business View Design Aid window:

Form	Description
Table Joins	Displays the tables that you select, along with the columns that are included in the tables. A key icon appears next to the primary key fields. The primary key fields are fields that are included in the primary index of the table. The primary table is where an application begins a search.
Available Tables	Enables you to locate tables and move them to the Table Joins form.
Selected Columns	Displays the data items that you select from the table to include in the business view.
Object Properties	Displays the properties of a data item that is selected in the Selected Columns form.
Available GT (Generic Text) Data Structures	Enables you to locate generic text data structures and move them to the Selected GT (Generic Text) Data Structures form. GT data structures are used for text search business views only.
Selected GT (Generic Text) Data Structures	Displays the generic text data structures that you select to include in the business view. GT data structures are used for text search business views only.

Consider these results when deleting business views and components of business views:

Deletion	Results
Deleting a data item from a business view.	If the data item is used in an application, you receive an error message when you attempt to run the application. If this occurs, you must open the application and delete the data item from the application.

Deletion	Results
Deleting an entire table from a business view.	You cannot run any of the applications that use the business view. If this occurs, you must open the application and delete all items that reside in the deleted table *or* attach a different business view to the application and reconnect all of the controls.
Deleting an entire business view.	Any forms that use the business view will fail. If this occurs, you must select a new business view for the forms and reconnect all of the controls.

7.3.2 Tables for Business Views

When creating custom business views, you must first select appropriate tables. The tables that you select must include the fields that are required to meet the business purpose. Although you can create one large business view to retrieve and update the columns that are required by the application, performance is negatively affected. Whenever possible, consider using table joins rather than creating one large business view that contains many data items.

Note: To ensure maximum performance in applications, use these guidelines for the number of tables that are joined in a business view:

Join five tables if all joins are simple joins.

Join three tables if any of the joins is an outer join or in the event of a table union.

See Also: ■Creating Table Joins.

7.3.3 Data Items for Business Views

If you include more than one table in a business view, you must indicate which table is the primary table. You can then select the required data items to include in the business view. All of the data items that reside in the tables that you selected are available for the business view.

Select the data items that are required by the interactive or batch application to include in the business view. When you create an application, you do not have to use every item in the business view. Balance between keeping the business view small for maximum performance and including enough fields to allow for future business requirements.

If the required data item appears in multiple tables, you typically want to select the data item from the primary table. Selecting the same data item from multiple tables causes the data item to appear multiple times in the business view. Each data item that is added to the business view includes its own table reference to identify its origin.

Note: If you include multiple tables in a joined business view, the primary key fields are automatically selected from the tables. You cannot remove the primary key fields from the business view.

If you include multiple tables in the business view, you must join the tables using a table join. You typically want to join tables on common key fields. You might also need to join on additional fields. You should join on as many fields as necessary to ensure that the data is fetched properly for each record. The joins must be performed on fields of the same field type.

Important: To ensure maximum system performance, do not include more than 256 columns in business views.

See Also: ▪Creating Table Joins.

7.3.4 Select Distinct

When a business view includes the primary key fields of the primary table, every row of the business view query is unique. If, for some reason, the business view does not contain all primary key fields of the primary table, then duplicate rows can occur during the business view query.

For example, Oracle's JD Edwards EnterpriseOne Journal Entry is unique by line number and document number. However, each document can contain multiple lines. If you need to display only one record per document, you can use Select Distinct to fetch only the first occurrence of the document number, not all of the detail lines within the document.

Business views that include a primary table that contains any of the following columns, which are used for currency support and security, might cause the Select Distinct feature to display duplicate values:

Column	Description
CO	Company
CRCD	Currency Code - From
CRDC	Currency Code - To
CRCX	Currency Code - Denominated In
CRCA	Currency Code - A/B Amounts
LT	Ledger Type
AID	Account ID
MCU	Business Unit
KCOO	Order Company (Company Code)
EMCU	Business Unit Header
MMCU	Branch
AN8	Address Number

7.3.4.1 Example: Select Distinct Feature

The business view that is used for this example, Event Detail Business View (V98EVDTL), uses the primary index of the primary table, F89EVDTL (Event Detail File), by default. The primary index of a JD Edwards EnterpriseOne table must be unique. A unique primary index ensures that the system does not return duplicate values when the business view query is generated. However, JD Edwards EnterpriseOne Business View Design Aid enables you to select any other index of the primary table when you process the business view.

These steps illustrate how the Select Distinct feature works:

1. In JD Edwards EnterpriseOne Business View Design Aid, select the primary table.
2. From the Table menu, select Change Index to change the primary table index.

   Change Index is available only for the primary table.

   The system displays a warning indicating that the selected column list will be changed.
3. Click Yes to continue.

   The Available Indices form appears. The first edit field on the form displays the current index of the table that is used by the business view. The default is the primary index.
4. For this example, select **Key by Formtyp, Evtype, Obj** from Available Indices, and click OK.

   The Table Joins form and the Selected Columns form reflect the keys of the new index.
5. Save the changes and quit JD Edwards EnterpriseOne Business View Design Aid.

   If you run an application that uses the V98EVDTL business view with Select Distinct disabled and the changed business view index (Key by Formtyp, Evtyp, Obj), the generated SQL statement is:

   ```
   SELECT EDOBJTYPE, EDEVTYPE, EDFORMTYPE FROM PVC. F98EVDTL
   ```

   Using this example, you might now have 281 rows of data from table F98EVDTL.
6. Reopen the V98EVDTL business view.
7. From the File menu, select Select Distinct.
8. Select Change Index to select the **Key by Formtyp, Evtyp, Obj** index from Available Indices, and then click OK.
9. Save the business view, and quit JD Edwards EnterpriseOne Business View Design Aid.

   You might need to quit the software and sign in again. The software stores the business view in cache memory. Even though you change a business view, the previous business view runs until it is cleared from cache.

   Generate and rerun the same application using the V98EVDTL business view with Select Distinct activated. The generated SQL statement is now:

   ```
   SELECT DISTINCT EDOBJTYPE, EDEVTYPE, EDFORMTYPE FROM PVC. F98EVDTL_Continue2
   ```

   Using this example, you might now have only 53 rows of data from table F98EVDTL.

7.3.5 Table Joins

Create table joins in business views to access data from multiple tables in a single application.

You typically use table joins for forms that are not input-capable, such as find/browse forms, and reports. You do not usually use joins for forms that update and add to the database. When you are updating the database, the relationship between the records must be precise. If you must use a table join for an input-capable form, only use a join where the relationship between the two tables is simple.

If a business view uses multiple tables, link the tables by establishing joins between columns in those tables. The links define how rows from one table correspond to rows in another table.

When you join a column in one table to a column in another table, both columns must be of the same data type. You can use the Object Properties form to view attributes for a column to determine whether you can use it in a join. The Object Properties form displays the attributes of the data item that you have highlighted on the Selected Columns form.

Review each table and determine how the data in one table is related to the data in the other tables. You might need to add columns or build new indices in a table, or even create new tables. If you build new indices, consider your business needs carefully before you do so.

> **Note:** To ensure maximum performance in applications, use these guidelines for the number of tables joined in a business view:
>
> Five tables if all joins are simple joins.
>
> Three tables if any of the joins is an outer join or in the event of a table union.

To join columns in a table join business view, use the mouse cursor to draw a line from a column in the primary table to the associated column in a secondary table. When you click the line that you drew, you can define the join by selecting a join type and an operator from the Join menu. These menu options are not available until you have clicked the line joining the columns.

The default join type is simple, and the default operator is equal. Available operators are:

- Equal (=)
- Less than (<)
- Greater than (>)
- Less than or equal (< =)
- Greater than or equal (> =)

7.3.6 Table Unions

Use table unions to pull rows from tables that have the same structure. Table unions pull rows that exist in both tables. The table union option is available only if the rows in one table are also included in the other table.

7.4 Creating Business Views

This section discusses how to:

- Add business views.
- Select tables for business views.
- Select data items for business views.
- Use Select Distinct.
- Create table joins.

- Create table unions.

7.4.1 Forms Used to Add Business Views

Form Name	FormID	Navigation	Usage
Object Management Workbench	W98220A	EnterpriseOne Life Cycle Tools, Application Development (GH902), Object Management, Object Management Workbench	Select and view objects that are included in projects, and access the JD Edwards EnterpriseOne design tools.
Add EnterpriseOne Object to the Project	W98220C	Click the Objects node of a project, and click Add on the Object Management Workbench form.	Add a new object to a project.
Add Object	W9861AF	Select Business View, and click OK on the Add EnterpriseOne Object to the Project form.	Add business views.
Business View Design Aid	W9860AL	Complete the object information, and click OK on the Add Object form.	Access Business View Design Aid, and create a type definition.

7.4.2 Adding Business Views

Access the Add Object form.

Object Name
Enter a unique name for the business view. Use the recommended naming convention for naming JD Edwards EnterpriseOne objects.

Description
Enter a meaningful description of the business view.

Product Code
Select a user-defined code (UDC) (98/SY) from the range of values that are reserved for clients, 55–59.

Product System Code
Select a UDC (98/SY) that represents the JD Edwards EnterpriseOne system where the data is used. This value is used for reporting and for description override purposes. Example values include:

01: Oracle's JD Edwards EnterpriseOne Address Book

03B: Oracle's JD Edwards EnterpriseOne Accounts Receivable

04: Oracle's JD Edwards EnterpriseOne Accounts Payable

09: Oracle's JD Edwards EnterpriseOne General Accounting

11: Oracle's JD Edwards EnterpriseOne Multicurrency

Object Use
Select a UDC (98/FU) that indicates the use of the object. For example, the object might be used to create a program, a master file, or a transaction journal. The value 300 represents business views.

Object Type
Displays an abbreviation that identifies the type of object that is being created. The object type for a business view is BSVW. This field is populated by the system based on the type of object that you selected on the Add Object form.

Text Search
Select this option to indicate that the business view is used by the runtime engine during a text search. When designing the business view, you must also add at least one GT (general text) data structure to the business view so that the runtime engine can use it in text searches.

7.4.3 Selecting Tables for Business Views

Access the JD Edwards Business View Design Aid form.

1. On the Design Tools tab, click Start the Business View Design Aid.
2. On the Available Tables form, use the query by example (QBE) line to search for an appropriate table.
3. Select one or more tables, and drag them to the Table Joins form.

   This form is called Table Joins regardless of whether you are joining multiple tables or working on a single table.
4. If more than one table is selected, double-click the title bar of the appropriate table to designate it as the primary table.

   If the business view contains multiple tables, the system automatically designates the first table that is added to the Table Joins form as the primary table. A crown icon appears in the upper-left corner of the primary table. If a business view contains only one table, that table is the primary table by default.

> **Note:** To delete a table from a business view, select the table and select Delete from the Table menu. You can also right-click the table and select Delete from the pop-up menu.

7.4.4 Selecting Data Items for Business Views

Access the JD Edwards Business View Design Aid form.

1. On the Table Joins form, double-click the data items to include in the business view.

   Selected data items appear with a check mark on the Table Joins form. As you select each data item, the system displays it on the Selected Columns form.
2. To remove data items from a business view, double-click the data item either on the Table Joins form or on the Selected Columns form.

7.4.5 Using Select Distinct

Access the JD Edwards Business View Design Aid form.

1. Select the primary table of the business view.

2. From the Table menu, select Distinct Mode.
3. From the Table menu, select Change Index.

   This option changes the index of the primary table to a non-unique index.

7.4.6 Creating Table Joins

Access the JD Edwards Business View Design Aid form.

1. On the Tables Joins form, click and draw a line that connects a column in the primary table to an associated column in a secondary table.

   Although the column names do not have to be the same, the attributes for Data Type and Decimals must be identical before you can create a table join between two columns. To determine whether data items are candidates for a join, click a data item on the Table Joins form and view the data item attributes that are displayed on the Object Properties form.
2. Click the line that joins the two columns.

   Both fields are highlighted.
3. From the Join menu, select Types, and then select an appropriate join type.

   Simple is the default join type.
4. From the Join menu, select Operators, and then select an appropriate operator.

   Equal is the default join operator.
5. To delete a join, click the line that connects the two fields, and then select Delete from the Join menu.

   You can also right-click the join and select Delete from the pop-up menu.

7.4.7 Creating Table Unions

Access the JD Edwards Business View Design Aid form.

1. From the Table menu, select Union Mode.

   You can also click the Union Mode button on the toolbar.

   The Union Mode feature is available only if all columns in one table also reside in the other table.
2. Select the tables for which you want to create a table union.

Part IV

Converting Tables

8 Understanding Table Conversion

This chapter contains the following topics:

8.1 Table Conversions

Table conversion is a type of batch process that enables you to rapidly manipulate the data in tables. This table describes the conversion types that the Oracle's JD Edwards EnterpriseOne Table Conversion tool uses to manipulate data:

Conversion Type	Description
Data Conversion	Enables you to transfer or copy data from an input table or business view into output tables using the logic that is necessary to perform the transfer. You can also use Data Conversion to update records in a table or business view.
Data Copy	Enables you to copy tables from one data source or environment to another data source or environment when the tables are identical.
Data Copy with Table Input	Enables you to copy tables based on information from an input table. For example, the input table might provide information about which tables are copied, where they are copied, and so on.
Batch Delete	Enables you to delete records from a table or business view.

The JD Edwards EnterpriseOne Table Conversion tool uses Oracle's JD Edwards EnterpriseOne tables, business views, and text files. The tool can also use non-JD Edwards EnterpriseOne tables, also referred to as *foreign tables,* as long as they reside in a database that is supported by JD Edwards EnterpriseOne software, such as:

- Oracle
- Access

- iSeries
- SQL Server

When creating a table conversion, you:

- Set up a table conversion template.
- Review the template.
- Create versions of the template.

  Like batch applications, table conversions include a template and versions. You can override some of the properties within a version at runtime.
- Process the table conversion.

  Table conversions can be saved and run multiple times. You can test the table conversion by running it in proof mode.

Oracle's JD Edwards EnterpriseOne Table Conversion Design Aid enables you to access any available environment for both input and output. The environments that you select determine which tables and business views are available for the conversion and where the tables reside. The environments that you select also determine the specifications, or descriptions, of tables and business views.

8.2 Types of Tables to Use

You can use these table types in table conversions:

Table Type	Description
JD Edwards EnterpriseOne tables	Tables that exist in the JD Edwards EnterpriseOne database and can be accessed from Oracle's JD Edwards EnterpriseOne Object Management Workbench. You can design and modify them using Oracle's JD Edwards EnterpriseOne Table Design Aid. At design time, only the specifications are needed to reference the table. At runtime, the table must be generated. An instance of the table must occur in a particular database.
Foreign tables	Tables that do not have a JD Edwards EnterpriseOne definition but reside in a database that is supported by JD Edwards EnterpriseOne. You must set up a data source and environment in JD Edwards EnterpriseOne to point to the location of foreign tables.

See Also:

- Preparing Foreign Tables for Table Conversion.

8.3 Business Views in Table Conversions

You must create a business view to establish a relationship among the input tables when you need to transfer data from:

- Multiple tables to a single table.
- Multiple tables to multiple tables.

A business view defines the relationship between two or more tables and joins the data into a single view. You can use JD Edwards EnterpriseOne business views for *input* to the table conversion only, not for output.

The system does not provide direct support for joining foreign tables. To use multiple foreign tables as input to a conversion, you must first define them through JD Edwards EnterpriseOne and then create a business view for them.

8.4 Text Files in Table Conversions

You can use table conversions to import directly from or export directly to a text file. When you convert a text file, it is stored with a single, long text field. User-defined formats are stored the same for a text file as for any table. With a text file conversion, the table name includes the path and the file name. If you do not specify the path with the file name, the default path is used.

8.5 Sort and Selection Criteria in Table Conversions

You can specify sort criteria in a table conversion. Sorting is used to process input rows in a sequence that groups related records together. The sort and selection features simplify the process of writing records to multiple tables in a typical one-to-many conversion. For example, if you have a single table of customer information, you can sort the data by area code and split the table into individual tables for each area code. Similarly, you can specify selection criteria for the input table to convert only a subset of that table.

JD Edwards EnterpriseOne Table Conversion Design Aid also enables you to add logic to determine when a change should occur to the value in a field.

8.6 Input and Output Environments

An Oracle's JD Edwards EnterpriseOne environment consists of:

- A path code.
- Oracle's JD Edwards EnterpriseOne Object Configuration Manager (OCM) mappings.

The path code of the environment is used to locate specification files for the environment. This path code is usually a subdirectory of the JD Edwards EnterpriseOne directory on a workstation. To reference JD Edwards EnterpriseOne tables in an environment, the full path code must exist on the machines where the conversion is created and run. Foreign tables can be referenced even if the path code does not exist.

OCM Mappings indicate where JD Edwards EnterpriseOne objects reside. JD Edwards EnterpriseOne Table Conversion Design Aid enables you to specify an input and output environment, which is used to locate input and output tables. To locate foreign tables, JD Edwards EnterpriseOne Table Conversion Design Aid uses the default OCM mapping for tables.

JD Edwards EnterpriseOne Table Conversion Design Aid uses these three environments when it processes a table conversion:

- The environment that you are signed into.

  Determines where the table conversion specifications are stored.
- The environment where the input tables reside.

- The environment where the output tables reside.

You can select one environment for the input tables (or business views) and one for the converted output tables. The three environments can be the same or they can be different.

When you use JD Edwards EnterpriseOne tables (or business views) in a table conversion, the environment provides the details of each table or business view, such as:

- Column names
- Data types
- Descriptions

Because this information comes from the JD Edwards EnterpriseOne specification tables, the system table does not have to exist in the database before you can design a table conversion; however, it must be created before the conversion is run. If you use a foreign table as input, you need to create it before you design the table conversion. The JD Edwards EnterpriseOne Table Conversion tool obtains its information about the table directly from the database where the foreign table resides. The environment also provides a default path for text files.

See Also:

- Designing Tables in JD Edwards EnterpriseOne.
- Preparing Foreign Tables for Table Conversion.
- "Setting Up Path Codes" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.
- "Setting Up Environments" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.
- "Setting Up Data Sources" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.

9 Setting Up Table Conversions

This chapter contains the following topics:

9.1 Understanding Table Conversion Setup

Oracle's JD Edwards EnterpriseOne Table Conversion Design Aid uses a director to guide you through the steps of setting up table conversion templates. Using the director, you can design table conversions for:

- Converting data.
- Copying tables between locations.
- Deleting records within tables.

You can also define user-defined formats for use as input or output tables in table conversion templates. User-defined formats are flat-file tables that store data as a continuous string of information, such as bank tapes.

After you set up a table conversion template, you can save it, create a version, and run the version multiple times.

See Also:

- Understanding User-Defined Formats.

9.2 Understanding the Table Conversion Process Flow

When you process a table conversion, the system triggers events that are similar to the events that are triggered when a report or application is run. These events are specific to the table conversion that you defined. Events provide pauses in the processing of the table conversion where you can attach logic.

In general, the event flow is the same for all table conversion types because these conversion types are all subsets of a data conversion:

- Data Copy.

  This conversion type does not include input and output tables; all actions are accomplished through the Process Begin event.

- Data Copy with Table Input.

  This conversion type does not include output tables; all actions are accomplished through the Process Begin, Process End, and Row Fetched events.

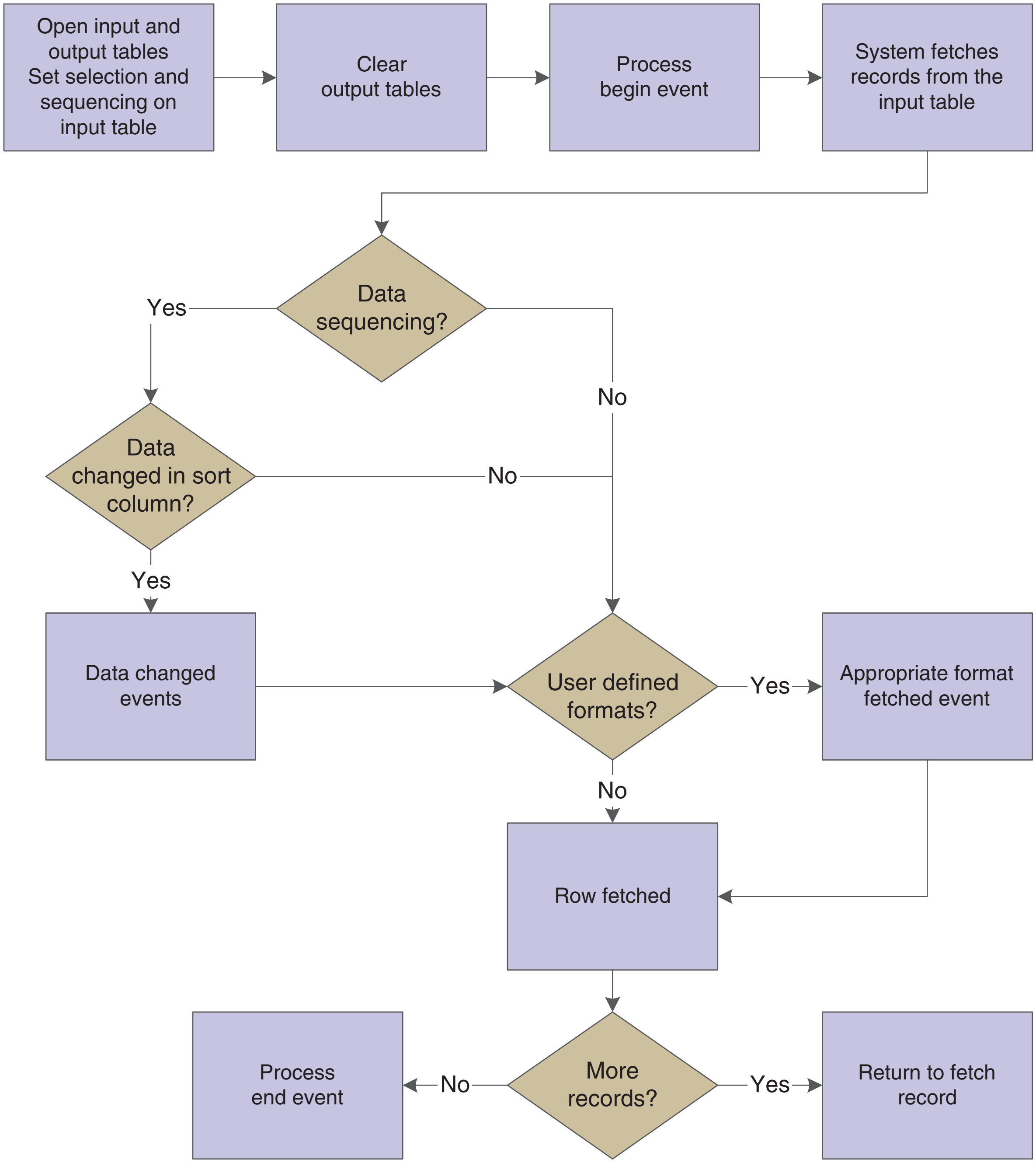
- Batch Delete.

  As with the Data Copy with Table Input type, this conversion type does not include output tables; all actions are accomplished through the Process Begin, Process End, and Row Fetched events.

The flexibility of events enables you to use table conversion types within other conversion types, if necessary.

Depending on the type of conversion that you define, some events might not be triggered. This diagram illustrates all events that can be triggered in a table conversion; events are indicated by *yes* or *no* decisions:

Figure 9–1 Table conversion event flow.



This table describes when each event occurs during the table conversion process:

Event	Description
Process Begin	Before fetching records from the input table, the system invokes the Process Begin event. At this point, you can attach logic that needs to run only once at the beginning of a table conversion, or attach any value that does not change for each record. This event is useful for mapping output fields that do not change for each record.

Event	Description
Data Changed	When you use data sequencing, the system invokes a Data Changed event for any sequenced field that changes. Data Changed events are not cascaded or hierarchical. For example, you can attach an event rule to this event to total a field or group of values.
Format Fetched	When you use user-defined formats (also known as flat files) in the input table, the system invokes a Format Fetched event for each record that is fetched from the input table. If you use multiple user-defined formats in a table conversion, the Format Fetched event is called for the particular format that is found in the record.
Row Fetched	An input table invokes a Row Fetched event after each row is fetched from the input table.
Process End	After the processing of all records is complete, the system invokes the Process End event. Attach event rules to the Process End event when you want the system to process logic after all input records have been read; for example, to write a total record to an output table or to write a record to a log file to record the status of the table conversion.

9.3 Adding Table Conversions

This section provides an overview of the navigation assistant, lists prerequisites, and discusses how to:

- Add new table conversions.
- Start the Table Conversion director.

9.3.1 Understanding the Navigation Assistant

The JD Edwards EnterpriseOne Table Conversion director includes a navigation assistant. The companion Navigation Assistant tracks where you are in the table conversion development process. You can right-click the Navigation Assistant to hide it for the current design process. The navigation assistant provides a visual indicator of the progress in designing the table conversion. Click a step on the navigation assistant to move forward or backward to another step in the process.

9.3.2 Prerequisites

Before you use the table conversion process, ensure that you:

- Set up a data source and environment for any foreign tables from which you might be importing data.

  See Preparing Foreign Tables for Table Conversion.
- Create a joined table business view if you are converting data from multiple tables.

  See Table Joins.
- Create a business function to perform validation if you want the data items within a table to validate against the data dictionary as part of the conversion process.

  The Table Conversion tool does not provide automatic data dictionary validation for inputs or outputs.

See "Using Business Functions" in the *JD Edwards EnterpriseOne Tools Development Tools: APIs and Business Functions Guide*.

9.3.3 Forms Used to Add New Table Conversions

Form Name	FormID	Navigation	Usage
Object Management Workbench	W98220A	EnterpriseOne Life Cycle Tools, Application Development (GH902), Object Management, Object Management Workbench	Select and view objects in projects, and access the JD Edwards EnterpriseOne design tools.
Add EnterpriseOne Object to the Project	W98220C	Click the Objects node of a project and click Add. on the Object Management Workbench form.	Add a new object to a project.
Add Object	W9861AF	Select Batch Application and click OK on the Add EnterpriseOne Object to the Project form.	Create a batch application object.
Batch Application Design	W9860AL	Complete the object information and click OK on the Add Object form.	Access the Table Conversion director.

9.3.4 Adding New Table Conversions

Access the Add Object form.

Object Name
Enter a unique name for the table conversion. Use the recommended naming convention for JD Edwards EnterpriseOne objects.

Description
Enter a meaningful description of the table conversion.

Product Code
Select a user-defined code (UDC) (98/SY) from the range of values that is reserved for clients, 55–59.

Product System Code
Select a UDC (98/SY) that represents the Oracle's JD Edwards EnterpriseOne system where the data is used. This value is used for reporting and description override purposes. Example values include:

01: Oracle's JD Edwards EnterpriseOne Address Book

03B: Oracle's JD Edwards EnterpriseOne Accounts Receivable

04: Oracle's JD Edwards EnterpriseOne Accounts Payable

09: Oracle's JD Edwards EnterpriseOne General Accounting

11: Oracle's JD Edwards EnterpriseOne Multicurrency

Object Use
Select a UDC (98/FU) that indicates the use of the object. For example, the object might be used to create a program, a master file, or a transaction journal. A 132 value represents a data file conversion.

Object Type
Displays an abbreviation that identifies the type of object that is being created. The object type for a table conversion is UBE. This field is populated by the system based on the type of object that you selected on the Add Object form.

Table Conversion
Select to enable JD Edwards EnterpriseOne Table Conversion Design Aid. If you do not select this option, Oracle's JD Edwards EnterpriseOne Report Design Aid is launched by default.

9.3.5 Starting the Table Conversion Director

Click OK on the Add Object form:

1. Select the Design Tools tab, and click Start Table Conversion Design Aid.

   The system displays the introduction form for the JD Edwards EnterpriseOne Table Conversion director.

2. Proceed to the task for the type of conversion that you want to perform.

See Also:

- Converting Data.
- Copying Data.
- Copying Data with Table Input.
- Deleting Groups of Records.

9.4 Converting Data

This section provides an overview of data conversion and discusses how to:

- Define external data for table conversions.
- Define input and output environments for table conversions.
- Define table conversion input.
- Define table conversion output.
- Map inputs to outputs.
- Specify table conversion logging options.
- Review table conversion results.

9.4.1 Understanding Data Conversion

Use the Data Conversion feature on the JD Edwards EnterpriseOne Table Conversion director to move data into tables from:

- A single table.
- Multiple tables that are defined in a business view.
- A single text file.

9.4.1.1 Data Conversion Example

This data conversion example illustrates how to copy employee records from the F0101 table in the Login Environment to the F0101 table in the PY812 environment.

1. On the Introduction form of the JD Edwards EnterpriseOne Table Conversion director, select Data Conversion, and click Next.
2. On the External Data form, accept the default selections, and click Next.
3. On the Select Environments form, select **<LOGIN ENV>** as the input and **PY812** as the output environment, and click Next.
4. On the Select Input form, drag the Address Book Master table to the Description column on the Table tab, and click Next.
5. On the Sequencing form, accept the default selection, and click Next.
6. On the Data Selection form, define the data selection as **Where AT1 (Search Type) (F0101) is equal to E** to select current employees, and click Next.
7. On the Select Outputs form, drag the Address Book Master table to the Description column on the Tables tab, and click Next.
8. On the Table Options form, clear the Run Currency Triggers option because the Address Book Master file does not contain currency information, and click Next.
9. On the Mapping form, select the Row Fetched event because you are converting data from similar files, click Map Same, and click Next.

   The system automatically maps data elements with the same aliases.
10. On the Logging Options form, select Log All Errors and Log Outputs, and click Next.

    These options enable you to review the conversion after it is complete.
11. On the Finish form, select No, I will create a version of this table conversion later, and click Finish.
12. On the Table Conversion form, click Save before quitting.

9.4.2 Forms Used to Convert Data

Form Name	FormID	Navigation	Usage
Object Management Workbench	W98220A	EnterpriseOne Life Cycle Tools, Application Development (GH902), Object Management, Object Management Workbench	Select and view objects in projects, and access the JD Edwards EnterpriseOne design tools.
Add EnterpriseOne Object to the Project	W98220C	Click the Objects node of a project and click Add. on the Object Management Workbench form.	Add a new object to a project.
Add Object	W9861AF	Select Batch Application and click OK on the Add EnterpriseOne Object to the Project form.	Create a batch application object.

Form Name	FormID	Navigation	Usage
Batch Application Design	W9860AL	Complete the object information and click OK on the Add Object form.	Access the Table Conversion director.

9.4.3 Defining External Data for Table Conversions

Access the JD Edwards EnterpriseOne Table Conversion Director.

1. Select Data Conversion, and click Next.
2. On the External Data form, click Select to attach a predefined processing option template to the table conversion.
3. On the Select Processing Option Template form, select an appropriate processing option template, and click OK.
4. On the External Data form, click Define to create a data structure.
5. On the Report Data Structure form, define the table conversion data structure, and click OK.

   Data structures contain a list of parameters that can be used to pass data into the table conversion when called through a report interconnect.
6. Click Next.

See Also:

- "Understanding Data Structures" in the *JD Edwards EnterpriseOne Tools Development Tools: Data Structure Design Guide*.

9.4.4 Defining Input and Output Environments for Table Conversions

Click Next on the Introduction form of the JD Edwards EnterpriseOne Table Conversion director to access the Select Environments form.

1. Select the input and output environments to use.

> **Note:** Select **<LOGIN ENV>** if you are creating table conversions for a user who might not have the same environments that you have. This option ensures that the table conversion uses the environment that the user is signed in to.

2. When creating a table conversion to be run in a different environment from the one where you create it, and the <LOGIN ENV> is not appropriate for the type of conversion that you are creating, select either Force Version to Override Input Environment or Force Version to Override Output Environment, and click Next.

   For example, if you create a conversion to be used by a client who does not have the environments that you have, use these options. When the conversion is invoked at the client site, the table conversion does not run until the user selects the appropriate environments in which to run it.

9.4.5 Defining Table Conversion Input

Click Next on the Select Environments form to access the Select Input form.

1. Select the appropriate tab based on whether the input is a table, a business view, a foreign table, or a text file.
2. Use the query by example (QBE) line to search for the appropriate table, business view, or foreign table.

   You can select only one table or one business view per table conversion. If the input consists of multiple tables, you must create a joined table business view.
3. Enter the name of the text file on the Text File tab, or use the Browse button to locate the file.

   If you want to rename the selected file, enter the new file name, and click Rename.

   For the iSeries, input text files are stored in the Integrated File System (IFS). Enter the path to the IFS before the file name.

   If you are using a text file, or if you need to define a format for a table or business view, select the User Defined Format option.

   Note: Text files are the only type of file that you can rename.

4. If the input is a table or a business view, drag it to the Description column.

   If you change a table or business view, the system warns you that deleting tables removes all mappings from the table conversion.
5. To delete an input file, select it and press Delete.
6. Click Next.
7. On the Sequencing form, click Data Sequencing to define data sequencing for a table or business view.

   If you specify a text file for input, you cannot define data sequencing or data selection for that file.

   Note: When you define data sequencing, you create new events that are available to you in the Mapping section of the director. One new event is created for each sequence column that you define. The event is called XXXX Data Changed, where XXXX is the column alias, for example, ALPH Data Changed. Each time the value in one of these columns changes from its previous value, the Data Changed event is invoked for that column. This event is similar to a level break in report templates, except that the Data Changed events are not related to each other. Invoking one does not invoke the others.

8. Click Next.
9. On the Data Selection form, click in the Operator column, and select an appropriate operator.

   On the Data Selection form, **Where** is the default value in the Operator column for the first set of criteria. For subsequent statements, **And** and **Or** become the available values for the Operator column.
10. Click the Left Operand column, and select from a list of available objects.

> **Note:** You can define selection criteria for database table columns only. User-defined format columns are not available because they do not exist in the database.

11. Click the Comparison column, and select one of these comparison operators:

    – is equal to.

    – is equal to or empty.

    – is greater than.

    – is greater than or equal to.

    – is less than.

    – is less than or equal to.

    – is not equal to.

12. Click the Right Operand column, and select from a list of available objects, special values, or variables.

    The options in this column depend on the selection that you made in the Comparison column. Some of these options might be available:

    – Blank

      Enters a blank (space) value.

    – Literal

      Enables you to enter specific values, as described in the next step.

    – Null

      Indicates that no value is associated with the field.

    – Zero

      Enters a value of 0.

    – IC

      Indicates input table columns.

    – RI

      Indicates values passed through a report interconnect to this table conversion.

    – PO

      Indicates processing option values for this table conversion.

    – SL

      Indicates system literals.

13. If you enter a literal in the Right Operand column, you can enter values on these tabs of the Single value form:

    – Single value

      Enables you to enter a single value. An example might be a value that indicates a specific company.

    – Range of values

Enables you to enter a range of values. An example might include companies from **00001** to **00060.** When you are using a range of values, only the *is equal to, is equal to or empty,* and *is not equal to* operators are valid.

– List of values

  Enables you to enter a list of values. Enter each value individually in the field, and click Add after each entry.

  An example list of values might include several user-defined codes for search types, such as **C** for customers, *E* for employees, and **V** for vendors. When you are using the list of values, only *is equal to, is equal to or empty,* and *is not equal to* are valid operators.

  To delete a value, select the value and click Delete.

14. To delete a line of criteria on the Data Selection form, select the row header and click Delete.
15. To change the order of the criteria, select the row header, and click the Up or Down button.
16. Click Next.

9.4.6 Defining Table Conversion Output

Click Next on the Select Input form to access the Select Outputs form.

1. Drag the table (or tables) that you want to use as outputs to the Description column, and click Next.
2. For text file conversions, from the Text File tab, select the file to use as the output, and click Use.

   If you want to rename the selected file, enter the new file name, and click Rename.

   For the iSeries, input text files are stored in the IFS. Enter the path to the IFS before the file name.

   If you are using a text file, or if you need to define a format for a table or business view, select the User Defined Format option.

   Note: Text files are the only type of file that you can rename.

3. Click Next.
4. To delete an output, select the row and press Delete.

   If you are using a text file, or if you need to define a user-defined format for a table or business view, click User Defined Format.

5. Click Next.
6. On the Table Options form, select from these options:

   – Run Currency Triggers

     Select this option if a JD Edwards EnterpriseOne table that is included in the table conversion contains currency triggers. If a table contains currency fields and you do not select this option, the system cannot determine where the decimal should be placed within a field. If you do not select the currency trigger option, and the source or destination fields are currency fields that are used in a calculation, you might receive unexpected results.

Do not select the Run Currency Triggers option if the input and output data sources are the same type (for example, Oracle, iSeries, or SQL Server) and no calculations are performed. Selecting this option results in slower performance.

You should not use currency triggers for an environment that has a different path code from the sign-in environment.

– Clear Output Tables

  Select this option to clear the output table before the table conversion runs.

– Force Row by Row Processing

  Select this option to test the table conversion or to ensure that the conversion always runs in row-by-row mode.

  You can test a conversion to ensure that the mapping logic is correct. In this case, specify the number of rows to process. The number of rows can be defined in the jde.ini under [TCEngine] or when you submit the table conversion.

  Select this option if you know that the values in the input table will produce duplicate keys in the output, and you want only the non-duplicate keys to be inserted.

  Selecting this option results in slower processing.

– Buffer Inserts To Output Tables

  Select this option to improve conversion performance if the conversion does not include any event rules to process insertion errors and you are processing row by row.

7. Click Next.

9.4.7 Mapping Inputs to Outputs

Click Next on the Select Outputs form to access the Mapping form.

1. Select the event on which mapping will occur.

   In most cases, you use either the Row Fetched event or Format Fetched event. Use the Format Fetched event if you are working with a user-defined format.

2. Click Advanced ER to include additional event rule logic in the table conversion.

   The Event Rules Design form appears for the event that you selected on the Mapping form.

3. Click Map Same to allow the system to map the inputs directly to the outputs.

   If the input and outputs share some of the same data, these fields are a direct map. For JD Edwards EnterpriseOne tables, the system maps fields by data dictionary item. For foreign tables, the system maps by column name.

4. Drag inputs to outputs to manually map the inputs to the outputs.

   Note: Click Delete to erase the mapping for a selected output. Click Delete All to erase the mapping for all outputs.

5. If multiple output files exist, select each file from the outputs list, and map the appropriate input columns to the appropriate output columns.

6. To define advanced output, double-click an output column.

   The Advanced Outputs form enables you to define literals, calculations, and other mappings without using Advanced ER. You can use an advanced input to add a literal value into a field. You can also define a calculation to populate an output field, such as adding two input fields together.

7. On the Advanced Outputs form, select one of these tabs, and add the appropriate input:

   – Available objects

     Select the appropriate object, and click Apply.

   – Literal

     Enter the appropriate value, and click Apply.

   – Defaults

     Select Use Dictionary Defaults, and click Apply.

     Select this option to use the default value in the data dictionary at runtime. If no default values exist in the data dictionary, the system displays a warning message.

   – Calculation

     Click Define Calculation, create a calculation in Expression Manager, and click Apply.

8. Click Close on the Advanced Outputs form.

9. On the Mapping form, select Issue a Write for this Event? to insert a row to the selected output after performing all column mappings for this event.

   When you select Issue a Write for this Event?, the system attaches the TC Insert Row event system function. This system function is automatically inserted at the end of the event rules. You cannot move it to another area. To specify when and where a row is inserted, attach the User Insert Row system function using Advanced ER, and move it to wherever it is needed.

10. Click Next.

See Also:

- Understanding the Table Conversion Process Flow.
- Using Event Rules in Table Conversions.

9.4.8 Specifying Table Conversion Logging Options

Click Next on the Mapping form to access the Logging Options form.

1. Select from these options:

   – Log All Errors.

     Select this option to log every error, regardless of the logging option for type of operation that failed.

   – Log Every Input Record.

     Select this option to log every input record in the table conversion log file.

   – Log Outputs.

     Select this option to log every output record in the table conversion log file.

- Log Deletes.

  Select this option to log every record that is deleted in the table conversion log file.

- Log Updates.

  Select this option to log every record that is updated in the table conversion log file.

- Log Copy Table Actions.

  Select this option to log the tables copied in the table conversion log file.

- Log Details of Copy Table Actions.

  Select this option to log all of the details of each Copy Table action that is performed.

- Run in Proof Mode.

  Select this option to run the conversion in proof mode. In proof mode, the conversion is simulated and all actions are logged, but no data is changed.

  Note: Proof mode is not an absolute proof mode. In some situations, the proof output might differ from the real output. If you insert the same record twice, for example, it may seem as though it worked in proof mode but, in reality, only one of the inserts works when you run the conversion in final mode.

2. Click Next.

9.4.9 Reviewing Table Conversion Results

Click Next on the Logging Options form to access the Finish form.

1. Select one of these options:

   - Yes, create a version of this table conversion.

     If you select yes, enter the version name.

   - No, I will create a version of this table conversion later.

2. Click Finish to complete the process.
3. On the Table Conversion form, select the Table Conversion Mappings form to review the mapping.
4. Make changes, as necessary, by selecting the appropriate option from the View menu, and then click Save.
5. To review the runtime conversion mode, select Check Conversion Mode from the Conversion menu.

   A dialog box provides you with details of the conversion mode that is defined.

6. From the Conversion menu, select Exit.

9.5 Defining File Formats for Table Conversion Input or Output

This section provides an overview of user-defined formats, lists a prerequisite, and discusses how to:

- Define delimited file formats with a single format.
- Define delimited file formats with multiple formats.
- Define fixed-width file formats with a single format.
- Define fixed-width file formats with multiple formats.

9.5.1 Understanding User-Defined Formats

JD Edwards EnterpriseOne Table Conversion Design Aid uses user-defined formats to handle fixed-width or comma separated value (CSV) files in a table or text file. These files are collectively known as flat files because they do not have relationships defined like relational database tables. Usually, flat files are text files that are stored on a workstation or server. They are used to import or export data from applications that have no other means of interaction. For example, you might need to share information between JD Edwards EnterpriseOne and an external application. If the external application does not support one of the same databases that JD Edwards EnterpriseOne supports, flat files might be the only way to transfer data between the two applications.

In a flat file, records are stored as a continuous string of information. The user-defined format provides instructions on how data is presented.

This table illustrates a single database record with five columns of data: Last, First, Addr, City, and Phone:

Last	First	Addr	City	Phone
Doe	John	123 Main	Anytown	5551234

This record can be formatted as a comma-delimited string:

```
Doe, John, 123 Main, Anytown, 5551234
```

This record can also be formatted as a fixed-width column format, in which each column is eight characters long.

9.5.1.1 Importing and Exporting Text Files

When you select a text file for input or output in a table conversion and do not specify a path, a default path is used. Table conversions that use the default path run on any platform. If an explicit path or iSeries library name is indicated for the file, then the file is created exactly as specified. Table conversions that are stored in this way might not work on other platforms, depending on the nature of the file system on each platform.

The default paths on non-iSeries platforms are:

- path code\Import\file name
- path code\Export\file name

You cannot specify a default path for the iSeries. Rather, the default is always the Import or the Export directory under the path code of the input or output environment. For example, if you are running a table conversion against the DV812 environment, the path in the file system might be \E812\DV812\import\myfile.txt.

If the conversion specifies a file name that includes anything other than the file name and extension, such as \mytextfiles\myfile.txt, the conversion attempts to open the file as specified.

9.5.1.2 Using User-Defined Formats as Input

If you use user-defined input formats, add event rules on the Format Fetched event. Without these event rules, the system ignores the format, and the data from the input table is never made available to the table conversion. If the table conversion does not require event rule logic, add comments in the Format Fetched event in Event Rules Design.

User-defined formats work with text files and tables.

Because the procedure for importing and exporting data is database-specific, consult a database administrator for details.

9.5.1.3 Using User-Defined Formats as Output

If you use user-defined output formats, add event rules on the Format Fetched event. Without these event rules, the system ignores the format, and the data from the output table is never made available to the table conversion. If the table conversion does not require event rule logic, add comments in the Format Fetched event in Event Rules Design.

User-defined formats work with text files and tables.

Because the procedure for importing and exporting data is database-specific, you should consult a database administrator for details.

9.5.2 Prerequisite

Before you begin defining delimited file formats, ensure that you have selected the input and output files, as appropriate.

9.5.3 Defining Delimited File Formats with a Single Format

From the JD Edwards EnterpriseOne Table Conversion director, access the Select Input form or the Select Outputs form.

1. Select User Defined Format, and then click Define Format(s).
2. On the User Defined Format - Type form, select the Delimited - Characters such as commas or tabs separate each column option.
3. Select the Single Format - All rows have the same format option, and click Next.
4. On the User Defined Format - Column Delimiter form, select the delimiter that separates the columns in the file:
   - Tab
   - Comma
   - Semicolon
   - Space
   - Other
5. Select the textual qualifier that is used to enclose a string of text:
   - None

     Select this option if no character is used to qualify text in this user-defined file.
   - Single Quotations

Select this option if a single quotation mark is used to qualify text in this user-defined file.

– Double Quotations

Select this option if a double quotation mark is used to qualify text in this user-defined file.

6. If the user-defined format incudes column headings in the first row, select **The first row contains column headers,** and click Next.
7. On the User Defined Format - Column Layout form, select a format from the list of available formats.

   You define the columns for the format so that the system can parse the information from the file.
8. For each column, click Add to define the column.
9. On the New Column Properties form, modify these fields as required, and click OK.
   – Name
   – Length
   – Type
10. To edit an existing column, select it, click Edit, and modify the properties on the Column Properties form.
11. To model the columns after an existing table, business view, or foreign table, click Model, select the appropriate tab, and then select the table or business view that you want to use as a model for the user-defined format.

> **Note:** You cannot model the columns after an existing object unless the layout of the two objects match.

12. Click OK, and then click Next.

    The system copies the format from the model that you selected and places it into the column layout grid.
13. On the User Defined Format - Finish form, click Finish.

9.5.4 Defining Delimited File Formats with Multiple Formats

From the JD Edwards EnterpriseOne Table Conversion director, access the Select Input form or the Select Outputs form.

1. Select User Defined Format, and then click Define Format(s).
2. On the User Defined Format - Type form, select the Delimited - Characters such as commas or tabs separate each column option.
3. Select the Multiple Formats - Rows are in two (2) or more formats option, and click Next.
4. On the User Defined Format - Column Delimiter form, select the delimiter that separates the columns in the file:
   – Tab
   – Comma

- Semicolon
- Space
- Other

5. Select the textual qualifier that is used to enclose a string of text, and click Next:

   - None

     Select this option if no character is used to qualify text in this user-defined file.

   - Single Quotations

     Select this option if a single quotation mark is used to qualify text in this user-defined file.

   - Double Quotations

     Select this option if a double quotation mark is used to qualify text in this user-defined file.

6. On the User Defined Format - Multiple Format Definition form, enter the number of formats that are included in the user-defined format.

7. To define the character length of the Form Designator Column, enter a value in the Length field, and click Next.

8. On the User Defined Format - Multiple Format Names form, name each format and define the values for each using the Name and Designator columns, and then click Next.

   The Designator name should describe the data in the user-defined formats.

   For example, suppose that a text file contains purchase order information. Lines in the table have information for a whole purchase order with the first field designated as POH; lines with a first field designated as POI contain information about individual items in the purchase order; and lines in the table with a first field designated as POT contain information about purchase order totals. In this scenario, you would enter POH as the designator of the first format, POI as the designator of the second format, and POT as the designator of the third format.

   Note: You can rename the columns for each format to help you remember the formats with which you are working. For example, you can rename the columns according to their function in the file, such as Header, Detail, and Total. These names appear in the Inputs list on the Mapping form of the table conversion.

9. On the User Defined Format - Column Layout form, select a format from the list of available formats.

   You define the columns for the format so that the system can parse the information from the file.

   Note: To move rows, including the Format Designator, select the row and drag it to a new location.

10. To add additional columns, click Add to define the column.

11. On the New Column Properties form, modify these fields as required, and click OK:

- Name
- Length
- Type

12. To edit an existing column, select it, click Edit, and change the properties on the Column Properties form.
13. To model the columns after an existing table, business view, or foreign table, click Model, select the appropriate tab, and then select the table or business view that you want to use as a model for the user-defined format.

    Note: You cannot model the columns after an existing object unless the layout of the two objects match.

14. Click OK, and then click Next.

    The system copies the format from the model that you selected and places it into the column layout grid.

15. On the User Defined Format - Finish form, click Finish.

9.5.5 Defining Fixed-Width File Formats with a Single Format

Access the Select Input form or the Select Outputs form.

1. Select User Defined Format. and then click Define Format(s).
2. On the User Defined Format - Type form, select the Fixed Width - Columns are aligned at specific positions with specific widths option.
3. Select the Single Format - All rows have the same format option, and click Next.
4. On the User Defined Format - Column Layout form, click Add to define the column layout.
5. On the New Column Properties form, modify these fields as required, and click OK:
    - Name
    - Start
    - Length
    - Type
6. To edit an existing column, select the column, click Edit, and change the properties on the Column Properties form.
7. To model the columns after an existing table, business view, or foreign table, click Model, select the appropriate tab, and then select the table or business view that you want to use as a model for the user-defined format.

    Note: You cannot model the columns after an existing object unless the layout of the two objects match.

8. Click OK, and then click Next.

    The system copies the format from the model that you selected, and places it into the column layout grid.

9. On the User Defined Format - Finish form, click Finish.

9.5.6 Defining Fixed-Width File Formats with Multiple Formats

Access the Select Input form or the Select Outputs form.

1. Select User Defined Format, and then click Define Format(s).
2. On the User Defined Format - Type form, select the Fixed Width - Columns are aligned at specific positions with specific widths option.
3. Select the Multiple Formats - Rows are in two (2) or more formats option, and click Next.
4. On the User Defined Format - Multiple Format Definition form, enter the number of formats that are included in the user-defined format, and click Next.

   To define the character length of the Designator column, complete the Length field.
5. On the User Defined Format - Multiple Format Names form, enter a name for each format in the Name column, and click Next.

   The Designator name should describe the data in the user-defined formats.

   For example, suppose that a text file contains purchase order information. Lines in the table have information for a whole purchase order with the first field designated as POH; lines with a first field designated as POI contain information about individual items in the purchase order; and lines in the table with a first field designated as POT contain information about purchase order totals. In this scenario, you would enter POH as the designator of the first format, POI as the designator of the second format, and POT as the designator of the third format.

   Note: You can rename the columns for each format to help you remember the formats with which you are working. For example, you can rename the columns according to their function in the file, such as Header, Detail, and Total. These names appear in the Inputs list on the Mapping form of the table conversion.

6. On the User Defined Format - Column Layout form, select a format from the list of available formats.

   You define the columns for the format so that the system can parse the information from the file.
7. To add additional columns, click Add to define the columns.
8. On the New Column Properties form, modify these fields as required, and click OK:
   - Name
   - Start
   - Length
   - Type
9. To edit an existing column, select the column, click Edit, and change the properties on the Column Properties form.

10. To model the columns after an existing table, business view, or foreign table, click Model, select the appropriate tab, and then select the table or business view that you want to use as a model for the user-defined format.

> **Note:** You cannot model the columns after an existing object unless the layout of the two objects match.

11. Click OK, and then click Next.

    The system copies the format from the model that you selected and places it into the column layout grid.

12. On the User Defined Format - Finish form, click Finish.

See Also:

- Reviewing Table Conversion Results.
- Using Event Rules in Table Conversions.
- Understanding Table Conversion Processing.

9.6 Copying Data

This section provides an overview of copying data, lists prerequisites, and discusses how to define data copy actions.

9.6.1 Understanding Copying Data

You use the Data Copy feature on the JD Edwards EnterpriseOne Table Conversion director to copy tables from one environment or data source to another. You can also import a copy table script to use in the conversion.

9.6.2 Prerequisites

Before you begin defining the input and output environments for copying data, ensure that you review these tasks:

- Defining External Data for Table Conversions.
- Defining Input and Output Environments for Table Conversions.
- Specifying Table Conversion Logging Options.
- Reviewing Table Conversion Results.

9.6.3 Defining Data Copy Actions

Access the JD Edwards EnterpriseOne Table Conversion Director.

1. Select Data Copy, and click Next.
2. On the External Data form, select a processing option template, define a data structure, as appropriate, and click Next.
3. On the Select Environments form, select an input environment and an output environment, and click Next.
4. On the Select Actions form, enter the name of a table in the Table field of the grid.

If you want to copy a single table, select **<Literal>,** and enter the name of that table on the Single Value tab. If you do not know the name of the table that you want to copy, use the <Find a Table> option.

When you enter the name of a table, the system automatically populates the remaining fields for you.

5. On the Select Actions form, modify these fields as necessary:

   – To Table

     Enter either the last table in a range of tables to be copied, or leave the field blank if you are copying a single table.

   – Source Type

     Select Data Source if the input and output sources are data sources. Select Environment if the input and output sources are environments.

     When you select Data Source, the system retrieves table definitions from the specifications in the sign-in environment.

     When you select Environment, the system uses the input and output environments to locate data and specifications for the tables. This enables the specifications to be different in the input and output environment, but the data is copied.

   – Input Source

     Select the data source or environment from which the input is read.

   – Output Source

     Select the data source or environment to which the output is written.

   – Create

     When you select **<If Table Exists>,** the system creates the table in the output and runs the conversion *only* if both the table specifications and the actual table exist in the input.

     When you select **<Yes>,** the system creates the table in the output. If the table already exists in the output, the system deletes it and creates a new table.

     When you select **<No>,** the system assumes that the table already exists in the output and does not create it.

   – Clear

     When you select **<If Table Exists>,** the system clears the table in the output *only* if it exists in the input.

     When you select **<Yes>,** the system deletes all rows in the output table before copying the table.

     When you select **<No>,** the output table is not cleared.

     Note: If you select **<No>** to not clear the output table, the conversion could result in key conflicts.

   – Copy

     When you select **<Yes>,** the system copies the data from the input table to the output table using Map Same.

When you select **<No>,** no data is copied.

- Owner ID
- Owner Pwd (owner password)

  If the data source requires an owner ID and password, enter them here. If you leave these fields blank, the system enters the ID and password of the sign-in user, or **<None>** if the data source does not have security.

6. To import an existing copy table script from another location, click Import, locate the file that you want to import, and click Open.

   The system adds an action for each copy table item in the copy table script.

7. Click Advanced ER to add event rule logic to the copy table process, and click Next.

   You can use event rules to write a custom copy table script.

8. On the Logging Options form, select the appropriate logging options, and click Next.

9. On the Finish form, select a version option, and click Finish.

See Also:

- Using Event Rules in Table Conversions.
- Reviewing Table Conversion Results.
- Running Table Conversions.

9.7 Copying Data with Table Input

This section provides an overview of how to copy data with table input, lists prerequisites, and discusses how to define data copy with table input actions.

9.7.1 Understanding How to Copy Data with Table Input

The Data Copy with Table Input feature is similar to the Data Copy option, except that it allows information for the process to come from an input table. The input table might provide information about which tables are copied, where they are copied, and so on. The Data Copy with Table Input feature enables you to select data.

For example, suppose that you create a table that includes:

- A table name
- The next backup date
- The backup frequency

You might populate this table with a list of tables to be archived and information specifying how often they are archived. You can then use the Data Copy with Table Input feature to select all rows in which the backup date is *less than or equal to* the current date, and calculate a new backup date.

9.7.2 Prerequisites

Before you begin defining the input and output environments for copying data, ensure that you review these tasks:

- Defining External Data for Table Conversions.

- Defining Input and Output Environments for Table Conversions.
- Defining Table Conversion Input.
- Defining Data Copy Actions.
- Specifying Table Conversion Logging Options.
- Reviewing Table Conversion Results.

9.7.3 Defining Data Copy with Table Input Actions

Access the JD Edwards EnterpriseOne Table Conversion Director.

1. Select Data Copy with Table Input, and click Next.
2. On the External Data form, select a processing option template, define a data structure, as appropriate, and click Next.
3. On the Select Environments form, select an input environment and an output environment, and click Next.
4. On the Select Input form, select a table, business view, or foreign table, and click Next.
5. On the Sequencing form, define data sequencing as appropriate, and click Next.
6. On the Data Selection form, define data selection as appropriate, and click Next.
7. On the Table Options form, select Run Currency Triggers, if appropriate, and click Next.

   Select this option if a JD Edwards EnterpriseOne table included in the table conversion contains currency triggers. If a table contains currency fields and you do not select this option, the system cannot determine where the decimal should be placed within a field. If you do not select the currency trigger option and the source or destination fields are currency fields used in a calculation, you might receive unexpected results.

   Do not select Run Currency Triggers if the input and output data sources are the same type (for example, Oracle, iSeries, or SQL Server) and no calculations are being performed. Selecting this option results in slower processing.

   You should not use currency triggers for an environment that has a different path code than the login environment.
8. On the Select Actions form, enter the name of a table in the Table field, modify the remaining fields as appropriate, and click Next.
9. To import an existing copy table script from another location, click Import, locate the file that you want to import, and click Open.
10. Click Advanced ER to add event rule logic to the copy table process, and click Next.
11. On the Logging Options form, select logging options, and click Next.
12. On the Finish form, select a version option, and click Finish.

9.8 Deleting Groups of Records

This section provides an overview of batch deletion, lists prerequisites, and discusses how to define batch deletion actions.

9.8.1 Understanding Batch Deletion

The Batch Delete feature enables you to delete a range of records from a JD Edwards EnterpriseOne input table or foreign table based on selection criteria that you define. For example, you can set up a batch delete table conversion that deletes records in an input table that does not contain valid data. You can also set up table conversions that delete all records from a particular table.

9.8.1.1 Example: Creating Purge Programs as a Batch Delete

This example illustrates how you can create a table conversion to delete records from the input environment. Designing purge programs as batch delete programs enables you to purge records with control and accuracy. You can archive the purged data or remove it from the system permanently. The archiving process is explained in this example.

Before you start this example, create a handle for the table.

See Understanding Handles.

1. On the introduction form of the JD Edwards EnterpriseOne Table Conversion director, select Batch Delete and click Next.
2. On the External Data form, select the **Purge Processing Option (T42000P)** processing option template, and click Next.
3. On the Select Environment form, select **<LOGIN ENV>** as the source environment, select Force Version to Override Input Environment, and click Next.

   The Force Version to Override Input Environment option prompts the person who runs the purge program to provide a valid source environment from which to run the batch-delete.
4. On the Select Input form, drag the **User Defined Code Types (F0004)** table to the Description column, and click Next.
5. On the Table Options form, select Run Currency Triggers, and click Next.
6. On the Data Selection form, define data selection as **Where IC UCD1 (User Defined Code - Class Code 1) is equal to Null,** and click Next.
7. On the Logging Options form, do not select any options, and click Next.
8. On the Finish form, select **Yes, create a version of this table conversion,** and enter **VER0001** as the version name.
9. On the warning form that indicates that the table conversion needs to be saved, click OK.
10. On the Selection for Batch Delete form, select the Process Begin event, and click Advanced ER.
11. In Event Rules Design, add these event rule variables with a report scope:
    - **FXXXXHandle_HFXXXX**
    - **szArchiveDataSource_DATS**
    - **szPurgeDataSource_DATS**
    - **szErrorCode_DTAI**
    - **cRenameFlag_EV01**
    - **mnErrorNumber_MATH01**

12. Enter these begin process event rules from R42119P, save the event rules, and quit Event Rules Design:

```
// Check to see if the purged data is being archived
If PO cArchive⇒
Records is equal to "1"
// If the environment processing option is blank, stop⇒
 processing.
If PO szArchiveEnvironmentName is equal to <Blank>
   Or PO sz⇒
ArchiveEnvironmentName is equal to <Null>
 Stop Conversion Processing("The⇒
 archive environment is invalid.")
Else
 // Check to make sure that the⇒
 archive environment and data source is not the
 // same as the input⇒
 environment and data source
 If PO szArchiveEnvironmentName is equal to SL⇒
 SourceEnvironment
  Stop Conversion Processing("The source and archive⇒
 environments are the same")
 Else
  //Get the data source of the archive⇒
 environment
Get and validate the data source for an environment/table ⇒
(B98700)
     PO szArchiveEnvironmentName -> szEnvironment
     "FXXXX" -> szTable⇒
Name (Replace x's with the name of your table.)
     VA rpt_szArchiveDataSource_⇒
DATS <- szDataSource
     VA rpt_szErrorCode_DTAI <- szErrorDataItem
     VA rpt_mn⇒
ErrorNumber_MATH01 <- mnErrorNumber
//SAR #4337575 - B98700 has been modified⇒
 to return an error if the table is
//not found in the data source. Since we⇒
 do not want the table to exist in the
//archive environment, we need to⇒
 bypass that error condition.
  If VA rpt_szErrorCode_DTAI is not equal to⇒
 <Blank>
    And VA rpt_szErrorCode_DTAI is not equal to <Null>
And VA rpt_⇒
szErrorCode_DTAI is not equal to "072W"
   Stop Conversion Processing("No⇒
 data source was found for the
archive environment")
  Else
   //Get the⇒
 data source of the source environment
Get and validate the data source for⇒
 an environment/table (B98700)
      SL SourceEnvironment -> szEnvironment
     ⇒
 "FXXXX" -> szTableName (Replace x's with the name of your table.)
      VA⇒
 rpt_szPurgeDataSource_DATS <- szDataSource
      VA rpt_szErrorCode_DTAI <- sz⇒
ErrorDataItem
```

```
      VA rpt_mnErrorNumber_MATH01 <- mnErrorNumber
0022If VA rpt_sz⇒
ErrorCode_DTAI is equalto "072W"
0023Stop Conversion Processing ("The table⇒
 could not be found
in the source environment.")
0024Else
0025If VA rpt_sz⇒
ErrorCode_DTAI is not equal to <Blank>
And VA rpt_szErrorCode_DTAI is not⇒
 equal to <Null>
0026Stop Conversion Processing ("No data source was found for
⇒
the source environment.")
0027End If
0028End If
0029    If VA rpt_szArchive⇒
DataSource_DATS is equal to VA rpt_szPurgeDataSource_DATS
0030    Stop Conversion⇒
 Processing ("The source and archive
environments have the same data source")
0031  ⇒
  Else
0032    // Open a table with the same table name in the output
environment.⇒
 The table
0033    // will be renamed later if the table name processing option was⇒
 populated.
0034    //Replace the X's in the following statement to reflect the
name⇒
 of your table.
0035    Copy Table Environment("FXXXX", <None>, SL Source⇒
Environment,
PO szArchiveEnvironmentName, <Yes>, <Yes>, <No>, <None>, <None>,⇒
 <Null>,
<Null>)
0036    // Open a handle to the archive table
0037    //⇒
 Replace the X's in the following statements to reflect the
name of your handle and⇒
 table.
0038    VA rpt_FXXXXHandle_HFXXXX = FXXXX.Open Handle
0039    If VA rpt_⇒
FXXXXHandle_HFXXXX is equal to <Null>
0040     Stop Conversion Processing ("Failed⇒
 to open FXXXX in the
archive environment")
0041    End If
0042    End If
0043   End If⇒
⇒
⇒
⇒
⇒
⇒
⇒
0044  End If
0045 End If
0046 End If
```

Note: Ensure that you map all event rule variables, even if you do not use every value.

In this example, the system writes log messages to the JDE.log and JDEDEBUG.log files on the Stop Conversion Processing event.

13. On the Selection for Batch Delete form, select the Row Fetched event, and click Advanced ER.
14. On Event Rules Design, enter these row fetched event rules from R42119P, save the event rules, and quit Event Rules Design:

```
// If we are archiving the purged records, write the record to the archive⇒
 table
//Replace the X's in the following statements to reflect the name of
⇒
your handle and table. Map the fields included in your table.
If PO cArchive⇒
Records is equal to "1"
FXXXX(VA rpt_FXXXXHandle_HFXXXX).Insert
   IC Order⇒
 Company(Order Number) -> TK Order Company(Order Number)
   IC Document(Order⇒
 No,Invoice,etc.) -> TK Document(Order No,Invoice,etc.)
   IC Order Type -> TK⇒
 Order Type
   IC Line Number -> TK Line Number
   IC Order Suffix -> TK Order Suffix
⇒
⇒
⇒
⇒
⇒
⇒
⇒
   IC Business Unit -> TK Business Unit
   IC Company -> TK Company
   IC Document⇒
 Company(Original Order) -> TK Document Company(Original Order)
   IC Original⇒
 Order Number -> TK Original Order Number
   IC Original Order Type -> TK Original⇒
 Order Type
   IC Original Line Number -> TK Original Line Number
   IC Company-Key ⇒
(Related Order) -> TK Company-Key (Related Order)
   IC Related PO/SO/WO Number ->⇒
 TK Related PO/SO/WO Number
   IC Related PO/SO/WO Order Type -> TK Related PO/SO⇒
/WO Order Type
   IC Related PO/SO Line Number -> TK Related PO/SO Line Number
   IC⇒
 Agreement Number-Distribution -> TK Agreement Number-Distribution
   IC Agreement⇒
 Supplement-Distribution -> TK Agreement Supplement-Distribution
   IC Address⇒
 Number -> TK Address Number
   IC Address Number-Ship To -> TK Address Number-Ship⇒
 To
   IC Address Number-Parent -> TK Address Number-Parent
   IC Date-Requested ->⇒
 TK Date-Requested
   IC Date-Order/Transaction -> TK Date-Order/Transaction
   IC⇒
 Date-Scheduled Pick -> TK Date-Scheduled Pick
   IC Date-Actual Ship Date -> TK⇒
 Date-Actual Ship Date
   IC Date-Invoice -> TK Date-Invoice
   IC Date-Cancel -> TK⇒
 Date-Cancel
   IC Date-For G/L(and Voucher) -> TK Date-For G/L(and Voucher)
   IC⇒
 Date-Promised Delivery -> TK Date-Promised Delivery
   IC Date-Price Effective⇒
 Date -> TK Date-Price Effective Date
   IC Date-Promised Shipment -> TK Date-⇒
Promised Shipment
   IC Reference -> TK Reference
   IC Reference 2 -> TK Reference⇒
 2
   IC Item Number-Short -> TK Item Number-Short
   IC 2nd Item Number -> TK 2nd⇒
 Item Number
   IC 3rd Item Number -> TK 3rd Item Number
   IC Location -> TK⇒
 Location
   IC Lot/Serial Number -> TK Lot/Serial Number
   IC From Grade -> TK⇒
 From Grade
   IC Thru Grade -> TK Thru Grade
   IC From Potency -> TK From Potency
 ⇒
  IC Thru Potency -> TK Thru Potency
   IC Days Before Expiration -> TK Days Before⇒
 Expiration
   IC Description -> TK Description
   IC Description-Line 2 -> TK⇒
 Description-Line 2
   IC Line Type -> TK Line Type
   IC Status Code-Next -> TK⇒
 Status Code-Next
   IC Status Code-Last -> TK Status Code-Last
   IC Business Unit -⇒
⇒
⇒
⇒
⇒
⇒
⇒
 Header -> TK Business Unit - Header
   IC Item Number - Related (Kit) -> TK Item⇒
 Number - Related (Kit)
   IC Kit Master Line Number -> TK Kit Master Line Number
  ⇒
 IC Component Line Number -> TK Component Line Number
   IC Related Kit Component -⇒
> TK Related Kit Component
   IC Number of Component Per Parent -> TK Number of⇒
 Component Per Parent
   IC Sales Catalog Section -> TK Sales Catalog Section
   IC⇒
 Sub Section -> TK Sub Section
   IC Sales Category Code 3 -> TK Sales Category⇒
 Code 3
   IC Sales Category Code 4 -> TK Sales Category Code 4
   IC Sales Category⇒
 Code 5 -> TK Sales Category Code 5
   IC Commodity Class -> TK Commodity Class
  ⇒
 IC Commodity Sub Class -> TK Commodity Sub Class
   IC Supplier Rebate Code -> TK⇒
 Supplier Rebate Code
   IC Master Planning Family -> TK Master Planning Family
  ⇒
 IC Purchasing Category Code 5 -> TK Purchasing Category Code 5
   IC Unit of⇒
 Measure as Input -> TK Unit of Measure as Input
   IC Units-Order/Transaction⇒
 Quantity -> TK Units-Order/Transaction Quantity
   IC Quantity Shipped -> TK⇒
 Quantity Shipped
   IC Units-Qty Backordered/Held -> TK Units-Qty Backordered/Held
⇒
   IC Units-Quantity Canceled/Scrapped -> TK Units-Quantity Canceled/Scrapped
   IC⇒
 Units-Future Quantity Committed -> TK Units-Future Quantity Committed
   IC Units-⇒
Open -> TK Units-Open
   IC Units-Shipped to Date -> TK Units-Shipped to Date
   IC⇒
 Units-Relieved -> TK Units-Relieved
   IC Committed (H/S) -> TK Committed (H/S)
  ⇒
 IC Other Quantity (1/2) -> TK Other Quantity (1/2)
   IC Amount-Price per Unit ->⇒
 TK Amount-Price per Unit
   IC Amount-Extended Price -> TK Amount-Extended Price
  ⇒
 IC Amount-Open -> TK Amount-Open
   IC Price Override Code -> TK Price Override⇒
 Code
   IC Temporary Price (Y/N) -> TK Temporary Price (Y/N)
   IC Unit of Measure-⇒
Entered for Unit Price -> TK Unit of Measure-
Entered for Unit Price
   IC Amount-⇒
List Price -> TK Amount-List Price
   IC Amount-Unit Cost -> TK Amount-Unit Cost
  ⇒
 IC Amount-Extended Cost -> TK Amount-Extended Cost
   IC Cost Override Code -> TK⇒
 Cost Override Code
   IC Extended Cost-Transfer -> TK Extended Cost-Transfer
   IC⇒
 Print Message -> TK Print Message
   IC Payment Terms Code -> TK Payment Terms Code
⇒
⇒
```

```
⇒
⇒
⇒
⇒
⇒
   IC Payment Instrument -> TK Payment Instrument
   IC Based on Date -> TK Based⇒
 on Date
   IC Discount-Trade -> TK Discount-Trade
   IC Trade Discount (Old) -> TK⇒
 Trade Discount (Old)
   IC Price and Adjustment Schedule -> TK Price and⇒
 Adjustment Schedule
   IC Item Price Group -> TK Item Price Group
   IC Pricing⇒
 Category Level -> TK Pricing Category Level
   IC Discount %-Cash -> TK Discount %-⇒
⇒
⇒
⇒
⇒
⇒
⇒
Cash
   IC Document Company -> TK Document Company
   IC Document⇒
(Voucher,Invoice,etc.) -> TK Document(Voucher,Invoice,etc.)
   IC Document Type ->⇒
 TK Document Type
   IC Document-Original -> TK Document-Original
   IC Document⇒
 Type-Original -> TK Document Type-Original
   IC Document Company-Original -> TK⇒
 Document Company-Original
   IC Pick Slip Number -> TK Pick Slip Number
   IC⇒
 Delivery Number -> TK Delivery Number
   IC Sales Taxable(Y/N) -> TK Sales Taxable⇒
(Y/N)
   IC Tax Rate/Area -> TK Tax Rate/Area
   IC Tax Expl Code 1 -> TK Tax Expl⇒
 Code 1
   IC Associated Text -> TK Associated Text
   IC Priority-Processing -> TK⇒
 Priority-Processing
   IC Printed Code -> TK Printed Code
   IC Backorders Allowed ⇒
(Y/N) -> TK Backorders Allowed (Y/N)
   IC Substitutes Allowed (Y/N) -> TK⇒
 Substitutes Allowed (Y/N)
   IC Partial Line Shipments Allowed (Y/N) -> TK Partial⇒
 Line Shipments Allowed (Y/N)
   IC Line of Business -> TK Line of Business
   IC⇒
 End Use -> TK End Use
   IC Duty Status -> TK Duty Status
   IC Nature of⇒
 Transaction -> TK Nature of Transaction
   IC Primary/Last Supplier Number -> TK⇒
 Primary/Last Supplier Number
```

```
   IC Carrier Number -> TK Carrier Number
   IC Mode⇒
 of Transport -> TK Mode of Transport
   IC Route Code -> TK Route Code
   IC Stop⇒
 Code -> TK Stop Code
   IC Zone Number -> TK Zone Number
   IC Container I.D. -> TK⇒
 Container I.D.
   IC Freight Handling Code -> TK Freight Handling Code
   IC⇒
 Shipping Commodity Class -> TK Shipping Commodity Class
   IC Shipping Conditions⇒
 Code -> TK Shipping Conditions Code
   IC Serial Number-Lot -> TK Serial Number-Lot
⇒
⇒
⇒
⇒
⇒
⇒
⇒
   IC Unit of Measure-Primary -> TK Unit of Measure-Primary
   IC Units-Primary⇒
 Quantity Ordered -> TK Units-Primary Quantity Ordered
   IC Unit of Measure-⇒
Secondary -> TK Unit of Measure-Secondary
   IC Units-Secondary Quantity Ordered -⇒
> TK Units-Secondary Quantity Ordered
   IC Unit of Measure-Pricing -> TK Unit⇒
 of Measure-Pricing
   IC Unit Weight -> TK Unit Weight
   IC Weight Unit of Measure⇒
 -> TK Weight Unit of Measure
   IC Unit Volume -> TK Unit Volume
   IC Volume Unit⇒
 of Measure -> TK Volume Unit of Measure
   IC Reprice (Basket Price) Category ->⇒
 TK Reprice (Basket Price) Category
   IC Order Reprice Category -> TK Order⇒
 Reprice Category
   IC Order Repriced Indicator -> TK Order Repriced Indicator
  ⇒
 IC Costing Method-Inventory -> TK Costing Method-Inventory
   IC G/L Offset -> TK⇒
 G/L Offset
   IC Century -> TK Century
   IC Fiscal Year -> TK Fiscal Year
   IC⇒
 Inter Branch Sales -> TK Inter Branch Sales
   IC On Hand Updated -> TK On Hand⇒
 Updated
   IC Configurator Print Flag -> TK Configurator Print Flag
   IC Sales⇒
 Order Status 04 -> TK Sales Order Status 04
   IC Substitute Item Indicator -> TK⇒
 Substitute Item Indicator
   IC Preference Commitment Indicator -> TK Preference⇒
 Commitment Indicator
   IC Ship date (PDDJ) overridden -> TK Ship date (PDDJ)⇒
 overridden
   IC Price Adjustment Line Indicator -> TK Price Adjustment Line⇒
 Indicator
   IC Price Adj. History Indicator -> TK Price Adj. History Indicator
  ⇒
 IC Preference Production Allocation -> TK Preference Production Allocation
   IC⇒
 Transfer/Direct Ship/Intercompany Flag -> TK Transfer/Direct Ship/
Intercompany⇒
 Flag
   IC Deferred entries flag -> TK Deferred entries flag
   IC Euro Conversion⇒
 Status Flag -> TK Euro Conversion Status Flag
   IC Sales Order Status 14 -> TK⇒
 Sales Order Status 14
   IC Sales Order Status 15 -> TK Sales Order Status 15
   IC⇒
 Apply Commission(Y/N) -> TK Apply Commission(Y/N)
   IC Commission Category -> TK⇒
 Commission Category
   IC Reason Code -> TK Reason Code
   IC Gross Weight -> TK⇒
 Gross Weight
   IC Gross Weight Unit of Measure -> TK Gross Weight Unit of Measure
⇒
   IC Subledger-G/L -> TK Subledger-G/L
   IC Subledger Type -> TK Subledger Type
  ⇒
 IC Code-Location Tax Status -> TK Code-Location Tax Status
   IC Price Code 1 ->⇒
 TK Price Code 1
   IC Price Code 2 -> TK Price Code 2
   IC Price Code 3 -> TK⇒
 Price Code 3
   IC Status-In Warehouse -> TK Status-In Warehouse
   IC Work Order⇒
 Freeze Code -> TK Work Order Freeze Code
   IC Send Method -> TK Send Method
   IC⇒
 Currency Code-From -> TK Currency Code-From
   IC Currency Conversion Rate-Spot⇒
 Rate -> TK Currency Conversion Rate-Spot Rate
   IC Amount-List Price per Unit ->⇒
 TK Amount-List Price per Unit
   IC Amount-Foreign Price per Unit -> TK Amount-⇒
Foreign Price per Unit
   IC Amount-Foreign Extended Price -> TK Amount-Foreign⇒
 Extended Price
   IC Amount-Foreign Unit Cost -> TK Amount-Foreign Unit Cost
   IC⇒
 Amount-Foreign Extended Cost -> TK Amount-Foreign Extended Cost
   IC User⇒
 Reserved Code -> TK User Reserved Code
   IC User Reserved Date -> TK User⇒
 Reserved Date
   IC User Reserved Amount -> TK User Reserved Amount
   IC User⇒
 Reserved Number -> TK User Reserved Number
   IC User Reserved Reference -> TK⇒
 User Reserved Reference
```

```
   IC Transaction Originator -> TK Transaction Originator
 ⇒
  IC User ID -> TK User ID
   IC Program ID -> TK Program ID
   IC Work Station ID -⇒
> TK Work Station ID
   IC Date-Updated -> TK Date-Updated
   IC Time of Day ->⇒
 TK Time of Day
   IC Manufacturing Variance Accounting Flag -> TK Manufacturing
⇒
Variance Accounting Flag
   IC Sales Order Status 17 -> TK Sales Order Status 17
  ⇒
 IC Sales Order Status 18 -> TK Sales Order Status 18
   IC Sales Order Status 19 -⇒
> TK Sales Order Status 19
   IC Sales Order Status 20 -> TK Sales Order Status⇒
 20
   IC Integration Reference 01 -> TK Integration Reference 01
   IC Integration⇒
 Reference 02 -> TK Integration Reference 02
   IC Integration Reference 03 -> TK⇒
 Integration Reference 03
   IC Integration Reference 04 -> TK Integration⇒
 Reference 04
   IC Integration Reference 05 -> TK Integration Reference 05
   IC⇒
 Source of Order -> TK Source of Order
   IC Reference -> TK Reference
   IC Demand⇒
 Unique Key ID -> TK Demand Unique Key ID
   IC Pull Signal -> TK Pull Signal
   IC⇒
 Release Number -> TK Release Number
   IC Scheduled Shipment Time -> TK Scheduled⇒
 Shipment Time
   IC Time-Rlease -> TK Time-Rlease
   IC Date-Rlease -> TK Date-⇒
Rlease
   IC Requested Delivery Time -> TK Requested Delivery Time
   IC Actual⇒
 Shipment Time -> TK Actual Shipment Time
   IC Time-Original Promised Delivery ->⇒
 TK Time-Original Promised Delivery
   IC Time-Scheduled Pick -> TK Time-Scheduled⇒
 Pick
   IC Time-Future Time 2 -> TK Time-Future Time 2
   IC Cross-Docking Flag ->⇒
 TK Cross-Docking Flag
   IC Cross-Docking Priority for Sales Orders -> TK Cross-⇒
Docking
Priority for Sales Orders
   IC Dual Unit of Measure Item -> TK Dual Unit⇒
 of Measure Item
   IC Buying Segment Code -> TK Buying Segment Code
   IC Current⇒
 Buying Segment Code -> TK Current Buying Segment Code
   IC Change Order Number ->⇒
 TK Change Order Number
```

```
   IC Address Number-Deliver To -> TK Address Number-⇒
Deliver To
   IC Pending Approval Flag -> TK Pending Approval Flag
   IC Revision⇒
 Reason -> TK Revision Reason
   IC Matrix Control Line Number -> TK Matrix⇒
 Control Line Number
   IC Shipment Number -> TK Shipment Number
   IC Promised⇒
 Delivery Time -> TK Promised Delivery Time
   IC Project Number -> TK Project⇒
 Number
   IC Sequence Number -> TK Sequence Number
   IC Item Revision Level -> TK⇒
 Item Revision Level
   IC Hold Orders Code -> TK Hold Orders Code
   IC Business⇒
 Unit-Header -> TK Business Unit-Header
   IC Business Unit-Demand -> TK Business⇒
 Unit-Demand
   IC Currency Code-Base -> TK Currency Code-Base
   IC Document Line⇒
 Number-Original -> TK Document Line Number-Original
   IC Date-Original Promised⇒
 Delivery -> TK Date-Original Promised Delivery
   IC Cross Dock Order Company⇒
(OrderNumber) -> TK Cross Dock Order
Company(OrderNumber)
   IC Cross Dock Order No -⇒
⇒
⇒
⇒
⇒
⇒
⇒
> TK Cross Dock Order No
   IC Cross Dock Order Type -> TK Cross Dock Order Type
  ⇒
 IC Cross Dock Line Number -> TK Cross Dock Line Number
   IC Cross Dock Order⇒
 Suffix -> TK Cross Dock Order Suffix
   IC Port of Entry or Exit -> TK Port of⇒
 Entry or Exit
   IC Payment Terms Override Code -> TK Payment Terms Override Code
 ⇒
  IC Buyer Number -> TK Buyer Number
   IC Promotion ID -> TK Promotion ID
   IC⇒
 Asset Item Number -> TK Asset Item Number
   IC Parent Number -> TK Parent Number
⇒
0005 // Do not delete the record if the insert to the archive table failed.
0006 If⇒
 SV Error_Status is not equal to CO ERROR
0007 Delete Current Input Row
0008 End⇒
 If
0009 Else
0010 Delete Current Input Row
0011 End If
```

Note: Ensure that you map all parameters to a field, even if you do not use every value.

15. On the Selection for Batch Delete form, select the Process End event, and click Advanced ER.
16. On Event Rules Design, enter these process end event rules from R42119P, save the event rules, and quit Event Rules Design:

```
0001 If PO cArchiveRecords is equal to "1"
0002 // Close the table
0003 //Replace⇒
 the X's in the following statement to reflect the
name of your handle and table.
⇒
0004 FXXXX(VA rpt_FXXXXHandle_HFXXXX).Close
0005 // If the data was archived and⇒
 the table name processing option was
populated,
0006 // rename the table.
0007 If PO⇒
 szArchiveTableName is not equal to <Blank>
  And PO szArchiveTableName is not⇒
 equal to <Null>
0008  Rename Table (B0000202)
   "FXXXX" -> szOldTableName
   PO sz⇒
ArchiveTableName -> szNewTableName
   "<Blank>" -> szTableOwnerID
   "<Blank>" -> sz⇒
Password
   VA rpt_szArchiveDataSource_DATS -> szDataSource
   VA rpt_cRenameFlag_⇒
EV01 <- cRenameTableSuccessful
0009 End If
0010 End If
```

Note: Ensure that you map all parameters to a field, even if you do not use every value.

See Also:

- Deleting Groups of Records.
- Understanding Handles.

9.8.2 Prerequisites

Before you begin defining the input and output environments for copying data, ensure that you review these tasks:

- Defining External Data for Table Conversions.
- Defining Input and Output Environments for Table Conversions.
- Defining Table Conversion Input.

- Defining Data Copy Actions.
- Specifying Table Conversion Logging Options.
- Reviewing Table Conversion Results.

9.8.3 Defining Batch Deletion Actions

Access the JD Edwards EnterpriseOne Table Conversion Director.

1. Select Batch Delete, and click Next.
2. On the External Data form, select a processing option template, define a data structure, as appropriate, and click Next.
3. On the Select Environments form, select a source environment, and click Next.
4. On the Select Input form, drag the appropriate table to the Description column, and click Next.

   To delete an input table, select it and press Delete.

   > **Note:** If you change the table, the system warns you that deleting tables removes all mappings from the table conversion.

5. On the Table Options form, select Run Currency Triggers, if appropriate, and click Next.

   Select this option if a JD Edwards EnterpriseOne table included in the table conversion contains currency triggers. If a table contains currency fields and you do not select this option, the system cannot determine where the decimal should be placed within a field. If you do not select the currency trigger option and the source or destination fields are currency fields used in a calculation, you might receive unexpected results.

   Do not select Run Currency Triggers if the input and output data sources are the same type (for example, Oracle, iSeries, or SQL Server) and no calculations are being performed. Selecting this option results in slower processing.

   You should not use currency triggers for an environment that has a path code that is different from the sign-in environment.

6. On the Data Selection form, define data selection as appropriate.
7. Select the **Row Fetched** event when deleting rows; otherwise, no records are deleted.

   When you run the table conversion, the system fetches the rows one at a time, runs the conversion for each row, and deletes the record from the input.

8. Click Advanced ER to insert additional event rule logic.
9. Verify that the Delete All Selected Records option is selected, and click Next.

   This option inserts the Delete Current Input Row system function into event rules.

10. On the Logging Options form, select logging options, as appropriate, and click Next.
11. On the Finish form, select a version option, and click Finish.

9.9 Using Event Rules in Table Conversions

This section provides an overview of event rules in table conversions and discusses how to create event rules in table conversions.

9.9.1 Understanding Event Rules in Table Conversions

You can use event rules to build complex functional logic into table conversions. For example, you can use event rules to insert information into a table or delete rows in a table based on specific conditions.

You attach event rules to an event, such as:

- Process Begin
- Row Fetched
- Format Fetched
- Process End

Event rules in table conversions include system functions that are specific to JD Edwards EnterpriseOne Table Conversion Design Aid.

9.9.1.1 Table Conversion System Functions

This table describes each system function that you can use in table conversion event rules:

System Function	Description
CopyTableDataSource	Enables you to copy a table or range of tables from one data source to another. The system copies tables based on specifications in the sign-in environment.
CopyTableEnvironment	Enables you to copy a table or range of tables from one environment to another. The system copies tables based on specifications in the input and output environments. If the specifications differ, the system performs a map-and-drop. This means that it creates a mapping between like fields in the source and destination tables and ignores all other fields.
TCInsertRow	The table conversion system inserts this system function when you select the Issue a write for this event? option and it cannot be moved. This function instructs the system that data should be written to the output table.
UserInsertRow	Enables you to specify when and where a row is inserted into the specified output table.
DeleteCurrentInputRow	Enables you to delete the current record from the input table.
UpdateCurrentInputRow	Enables you to update the current record in the input table after it has been modified.
SetSelectionAppendFlag	Enables you to determine whether selection criteria that is added by the system function SetUserSelection on the input table should be appended to the existing selection criteria or if it should replace the existing selection criteria.

System Function	Description
SetUserSelection	Enables you to conditionally modify data selection on the input table. Call SetSelectionAppendFlag before calling SetUserSelection to indicate whether to replace or append to the existing data selection.

See Also:

- Understanding the Table Conversion Process Flow.
- "Understanding Events, Event Rules, and Runtime Processing" in the *JD Edwards EnterpriseOne Tools Development Tools: Event Rules Guide*.

9.9.2 Creating Event Rules in Table Conversions

Access a table conversion form that includes an Advanced ER button.

1. From the Events drop-down list, select the event to which you want to attach event rules, and click Advanced ER.
2. On the Event Rules Design form, select any of these options to define specific business logic:
   - Assignment

     Select this option to assign a fixed value, a field, or a mathematical expression to a field or variable.
   - If\While

     Select this option to create If and While logic statements, which are conditional instructions for event rule logic.
   - Business Function

     Select this option to attach an existing JD Edwards EnterpriseOne business function to an event. Business functions can retrieve a next number value for a new customer or convert Julian calendar dates to month, day, and year.
   - System Function

     Select this option to attach an existing JD Edwards EnterpriseOne system function, such as Copy Table Environment or User Insert Row.
   - Variables

     Select this option to create event rule variables, using existing data items, to store values for additional processing. Event rule variables can be used to accumulate totals, record the number of records that are read, and so on.
   - Else

     Select this option to create Else logic statements. An Else statement is automatically inserted after an If statement.
   - Table I/O

     Select this option to open tables in the input, output, or sign-in environment. Table I/O enables you to read data from tables other than the input table and use the data to create an output record. For example, suppose that you create a table conversion that loops through records in the F0101 table. The table conversion copies the records to another table, loops back through the records to find each customer that has a specific employee as a contact, and copies the information to the output table.

- Report Interconnect

  Select this option to connect a batch process or report to the table conversion.

3. Define the logic, save the event rules, and quit Event Rules Design.

See Also:

- Working with Table I/O.
- "Using Event Rules Design" in the *JD Edwards EnterpriseOne Tools Development Tools: Event Rules Guide*.
- "Using Business Functions" in the *JD Edwards EnterpriseOne Tools Development Tools: APIs and Business Functions Guide*.

10

Running Table Conversions

This chapter contains the following topics:

10.1 Understanding Table Conversion Processing

You run a table conversion by submitting an associated batch version.

To track the table conversion process, you can use the tracing feature. The tracing feature writes the details of the table conversion process to a log. You can define the trace level to control the level of detail that is included in the log. When you test a table conversion, you can select to process the conversion one row at a time, which enables you to isolate problems or unexpected results.

10.2 Submitting Table Conversions

This section provides overviews of submitting table conversions and overriding table conversion properties and discusses how to:

- Override table conversion properties at runtime.
- Override table conversion properties of new versions.
- Override table conversion locations.
- Run table conversions on the web client.

10.2.1 Understanding Submitting Table Conversions

When you submit versions of table conversions, you can override table conversion properties, such as:

- Input and output environments.
- Trace level.
- Location where the table conversion is processed.

  Overriding the location where you process the table conversion enables you to process the table conversion even if the server that you normally use is inoperable.

See Also:

- "Submitting Table Conversions from the Microsoft Windows Client" in the *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.
- "Accessing Properties for Versions of Table Conversions" in the *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

10.2.2 Understanding Overriding Table Conversion Properties

When you submit versions of table conversions, you can override these properties:

Properties	Description
Environments	Enables you to change the location of the input or output data.
Data selection	Enables you to limit the number of input records.
Table options	Enables you to control how the JD Edwards EnterpriseOne Table Conversion system processes records during the conversion.
Logging options	Enables you to modify logging options.
Debug logging	Enables you to override the setting for debug logging as defined in the jde.ini.

10.2.2.1 Table Conversions on the Web Client

Submitting versions of table conversions from the web client is different from submitting them from the Microsoft Windows client. From the web client, you cannot modify properties. You can select and modify processing options and you can select these options:

- Override Location
- Override Job Queue
- Submit Version Specifications Only
- Data Selection

See "Copying Batch Version Specifications to an Enterprise Server" in the *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

10.2.3 Forms Used to Submit Table Conversions

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Batch Versions	Enter a table conversion template, and select a version.
Table Conversion Prompting	W98305D	Highlight a version on the Work With Batch Versions form, and click Select.	Work with versions.

Form Name	FormID	Navigation	Usage
Properties	NA	Select the Properties option on the Table Conversion Prompting form, and click Submit.	Review and override the environments, data selection, table options, and logging options that are specified within the conversion.
JD Edwards Data Sources	W98305B	Select the Override Location option on the Table Conversion Prompting form, and click Submit.	Select a data source to use as an override location.

10.2.4 Overriding Table Conversion Properties at Runtime

Access the Work With Batch Versions - Available Versions form.

> **Note:** When you override table conversion properties at runtime, the overrides are for that single process only; they are not saved with the version.

1. Enter the name of the table conversion template in the Batch Application field, and click Find.

   The name of the Batch Application field changes to Table Conversion.

   You can also use the visual assist in the Batch Application field to search for the correct table conversion.

2. Select a batch version, and click Select.
3. On the Table Conversion Prompting form, select Properties, and click Submit.
4. On the Properties form, review and override the environments, data selection, table options, and logging options that specified in the table conversion.

   These forms are similar to the forms in Table Conversion Design Aid.

5. Select the Debug Logging tab to perform these functions:

   – Select the Use ini settings for trace level and number of rows to process option to use the jde.ini settings for the trace level and row-by-row conversion process.

     This option ensures that the system uses the settings that are defined in the jde.ini instead of the values that are entered in the Trace Level and Number of Rows fields.

   – Clear the Use ini settings for trace level and number of rows to process option to override the trace level in the jde.ini. Enter a value from **0** to **10** in the Trace Level field.

   – Clear the Use ini settings for trace level and number of rows to process option to convert a specific number of records (for example, if you want to test the table conversion). Enter the number of rows to process in the Number of Rows field.

     When you enter **0** in the Number of Rows field, the system processes all rows.

This option corresponds to the StopAfterRow setting in the jde.ini file. If you enter a value here, you override any specifications in the jde.ini.

6. Click OK to save the changes to the version.

   The system processes the table conversion.

   See Also:

   - Testing Table Conversions.

10.2.5 Overriding Table Conversion Properties of New Versions

Access the Work With Batch Versions - Available Versions form.

> **Note:** If you override table conversion properties from the Tools tab, the new properties are saved with the version.

1. Enter the name of the table conversion in the Batch Application field, and click Find.

   The name of the Batch Application field changes to Table Conversion.

   You can also use the visual assist in the Batch Application field to search for the correct table conversion.

2. Add a new version of the table conversion, complete the version information, and click OK.

3. On the Table Conversion Version Design form, select the Tools tab, and click Properties.

4. On the Properties form, review and override the environments, data selection, table options, and logging options that are specified in the table conversion.

   These forms are similar to the forms in Table Conversion Design Aid.

5. To use debug logging, select the Debug Logging tab to perform these functions:

   – Select the Use ini settings for trace level and number of rows to process option to use the jde.ini settings for the trace level and row-by-row conversion process.

     This option ensures that the system uses the settings that are defined in the jde.ini instead of the values that are entered in the Trace Level and Number of Rows fields.

   – Clear the Use ini settings for trace level and number of rows to process option to override the trace level in the jde.ini. Enter a value from **0** to **10** in the Trace Level field.

   – Clear the Use ini settings for trace level and number of rows to process option to convert a specific number of records (for example, if you want to test the table conversion). Enter the number of rows to process in the Number of Rows field.

     When you enter **0** in the Number of Rows field, the system processes all rows.

     This option corresponds to the StopAfterRow setting in the jde.ini file. If you enter a value here, you override any specifications in the jde.ini file.

10.2.6 Overriding Table Conversion Locations

Access the Work With Batch Versions - Available Versions form.

1. Enter the name of the table conversion in the Batch Application field, and click Find.

   The name of the Batch Application field changes to Table Conversion.

   You can also use the visual assist in the Batch Application field to search for the correct table conversion.

2. Select a batch version, and click Select.
3. On the Table Conversion Prompting form, select Override Location, and click Submit.
4. On the JD Edwards Data Sources form, select the data source to use, and click Select.

10.2.7 Running Table Conversions from the Web Client

From EnterpriseOne Menu, select Submit Job to access the Submit Job-Work With Batch Versions-Available Versions form.

1. Enter the name of the table conversion in the Batch Application field, and click Find.

   The name of the Batch Application field changes to Table Conversion.

2. Select a batch version, and click Select.
3. On the Table Conversion Prompting form, select from these override options, and click Submit:
   - Override Location
   - Override Job Queue
   - Submit Version Specifications Only
   - Data Selection
4. Modify the appropriate property, and submit the batch version.

   See Also:

   - "Submitting Table Conversions from the Web Client" in the *JD Edwards EnterpriseOne Tools Development Tools: Batch Versions Guide*.

10.3 Testing Table Conversions

This section provides an overview of table conversion testing and discusses how to:

- Set the trace level for debug logging on the workstation.
- Set the trace level for debug logging on the server.
- Force row-by-row conversion.

10.3.1 Understanding Table Conversion Testing

You can test table conversions in proof mode to ensure that they are processed as expected without errors. You can log debug information about the conversion while it is being processed. You can also force the conversion to process one row at a time, which is useful if the conversion normally runs as an insert-from-select.

To log debug information for table conversions, enable tracing and set a trace level in the jde.ini. Use a value in the range of 0 to 10 for the trace level. These values indicate the level of detail that you want to include in the log. If necessary, you can override the jde.ini settings on the batch version of a table conversion.

If you set the trace level for logging at 1, the system logs basic information about the table conversion, such as:

- Name
- Inputs
- Outputs
- Event rule logic
- Number of rows inserted

If you set the trace level at 10, the system logs all information about every column in every format, including:

- User-defined formats.
- Processing options that are associated with the table conversion.
- All other information that is involved in the table conversion process.

The higher that you set the trace level, the more details the system provides regarding the table conversion process.

You can force row-by-row processing on the workstation or on the server when you want to test the table conversion. You can also set a trace level on the workstation or on the server, depending on where you process the table conversion. You can specify the number of rows to process in combination with forcing a row-by-row conversion.

10.3.1.1 Difference Between Logging Options and Debug Logging

Logging options, which you define when you set up a table conversion, can log:

- All errors that occur during the table conversion.
- All records that are copied, deleted, or updated.
- Details of copy table actions.

Debug logging logs more detailed information about the table conversion. This information enables you to determine the exact area in the table conversion where errors occurred.

> **Note:** If you modify the debug logging in the batch version, you override the settings in the jde.ini.

10.3.1.2 Trace Levels

You can define a trace level for debug logging either on a workstation or on a server, depending on where you run the table conversion.

This table describes the logging information for each trace level:

Trace Level	Logging Information
Level 1	Logs general information about the conversion, such as name, inputs, outputs, event rule logic, and how many rows were inserted.
Level 2	Logs function call traces, such as starting conversion, ending conversion, and inserting rows. Also includes all level 1 information.
Level 3	Logs the points at which event rules are processed. Also includes all level 1 and level 2 information.
Level 4	Not applicable.
Level 5	Logs the points at which jdeCallObject is processed, such as calls to business functions from event rules. Also includes all level 1, level 2, and level 3 information.
Levels 6-8	Not applicable.
Level 9	Logs the content of columns during input, the processing of event rules, and the content before output. Also includes all level 1, level 2, level 3, and level 5 information.
Level 10	Logs all information for all applicable levels.

Important: Do not set the trace level at 10 when running a table conversion on tables that contain large amounts of data. The system writes the data to the server, and large amounts of data can cause the server to run out of disk space. You can, however, specify the number of rows to process on the batch version properties form.

10.3.2 Setting the Trace Level for Debug Logging on the Workstation

To define the trace level for debug logging on the workstation.

1. Open the jde.ini on the workstation.
2. To enable tracing and to set the trace level on a workstation, *add* this new information to the [TCEngine] section of the jde.ini:

   ```
   [TCEngine]
   _continue-
   TraceLevel=n
   ```

   where n is a number from 0 through 10.
3. To enable tracing and to set the trace level on a workstation, *modify* this information in the [Debug] and [UBE] sections of the jde.ini:

   ```
   [Debug]
   Output=File

   [UBE]
   UBESaveLogFile=1
   ```

10.3.3 Setting the Trace Level for Debug Logging on the Server

To define the trace level for debug logging on the server.

1. To enable tracing and to set the trace level on a server, set the following information in the [TCEngine] section of the server jde.ini:

   ```
   [TCEngine]

   TraceLevel=n
   ```

   where n is a number from 0 to 10.

2. To enable tracing and to set the trace level on a server, *modify* this information in the [Debug] section of the server jde.ini:

   ```
   [Debug]
   Output=File

   KeepLogs=1
   ```

10.3.4 Forcing Row-By-Row Conversion

To force row-by-row conversion and specify the number of rows to process:

1. Open the jde.ini file.

2. To force row-by-row conversion, add this *new* information to the [TCEngine] section of the jde.ini:

   ```
   [TCEngine]
   ForceRowByRow=1
   _Continue -
   ```

3. To specify the number of rows to process, add this *new* information to the [TCEngine] section of the jde.ini:

   ```
   StopAfterRow=n
   ```

   where n is the number of rows that you want to process.

11

Preparing Foreign Tables for Table Conversion

This chapter contains the following topics:

- Section 11.1, "Understanding Foreign Tables"
- Section 11.2, "Adding Data Sources"
- Section 11.3, "Adding Environments"
- Section 11.4, "Setting Up Default OCM Mappings"

11.1 Understanding Foreign Tables

Foreign tables are text files, or any other files or tables, that are not recognized by Oracle's JD Edwards EnterpriseOne software. However, they must reside in a database that is supported by JD Edwards EnterpriseOne, such as:

- Oracle
- Access
- iSeries
- SQL Server

Before you can work with foreign tables in Oracle's JD Edwards EnterpriseOne Table Conversion Design Aid, you must define the database to JD Edwards EnterpriseOne. This definition is created when you set up an ODBC data source or an Oracle Call Interface (OCI) data source that points to the database where the foreign tables reside.

You must also add a data source in JD Edwards EnterpriseOne that points to the ODBC or OCI data source that you defined. For each ODBC data source, database instance, or library that contains foreign tables, you must set up an environment. The environment points to the JD Edwards EnterpriseOne data source, which in turn points to the database or library. In addition, you must set up an ODBC mapping from the data source to the environment.

Note: When you work with foreign tables, you must discuss database permissions with a database administrator. Your JD Edwards EnterpriseOne user ID (or, if you are using the JD Edwards EnterpriseOne security server feature, the database user to which it maps) might need to be changed to give you permissions to use the tables in the foreign database. Without these permissions, you cannot view the foreign tables in JD Edwards EnterpriseOne Table Conversion Design Aid. Under certain conditions, the table conversion engine must create temporary tables in the output environment and requires create-and-drop permissions for the database.

11.2 Adding Data Sources

This section discusses how to add Oracle's JD Edwards EnterpriseOne data sources.

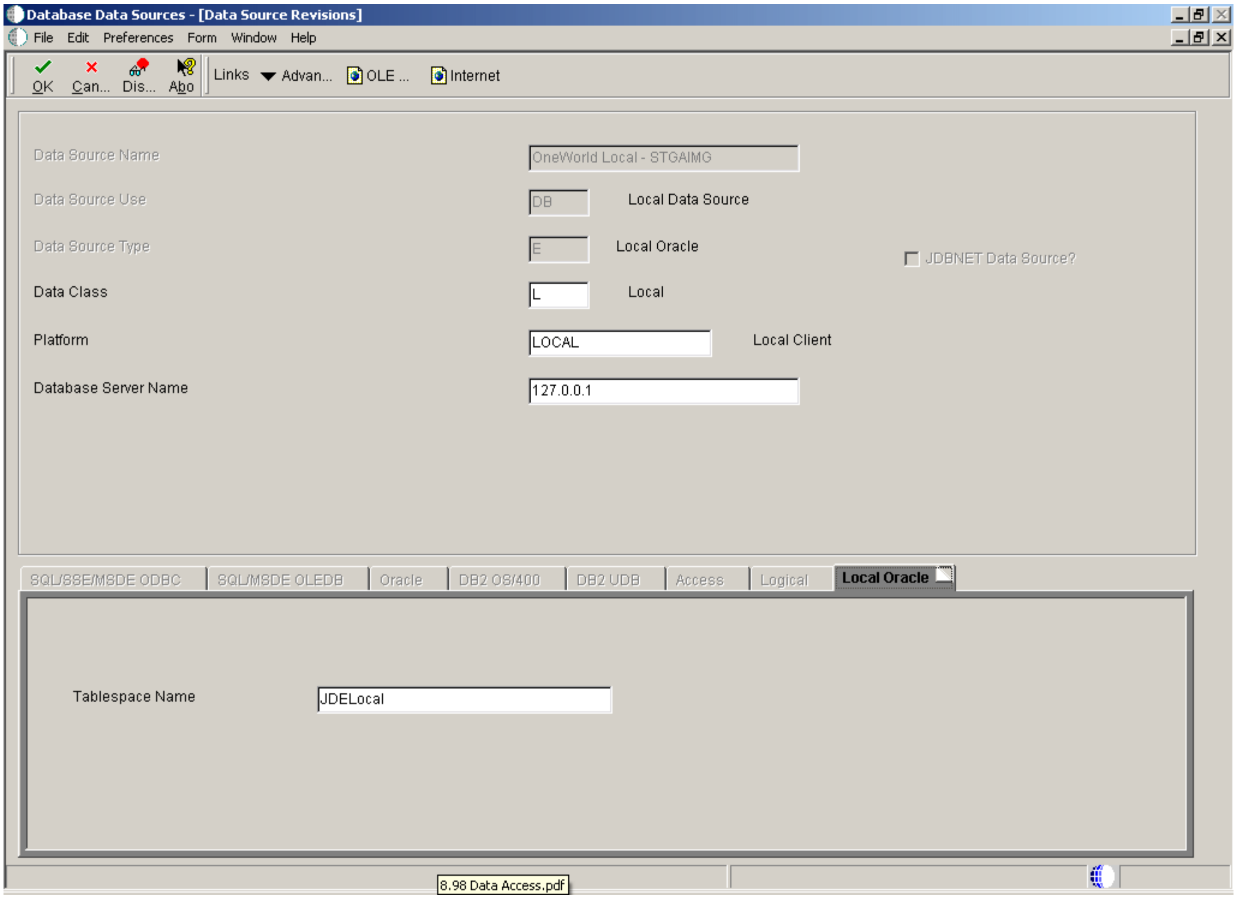
11.2.1 Forms Used to Add Data Sources

Form Name	FormID	Navigation	Usage
Machine Search & Select	W986115E	EnterpriseOne Life Cycle Tools, System Administration Tools, System Administration Tools (GH9011), Data Source Management, Database Data Sources	Select where the data source resides.
Work With Data Sources	W986115A	Select the appropriate machine on which the data source resides, and then click Select on the Machine Search & Select form.	Add a data source, or select a data source to modify.
Data Source Revisions	W986115O	Click Add on the Work With Data Sources form.	Add JD Edwards EnterpriseOne data sources. Enter data source name, use, type, data class, platform, server name, database name, and owner.

11.2.2 Adding Data Sources

Access the Data Source Revisions form.

Figure 11–1 Data Source Revisions form



Data Source Name	OneWorld Local - STGAIMG	
Data Source Use	DB	Local Data Source
Data Source Type	E	Local Oracle
Data Class	L	Local
Platform	LOCAL	Local Client
Database Server Name	127.0.0.1	

□ JDBNET Data Source?

Tablespace Name: JDELocal

Data Source Name

Enter a unique name that identifies the data source.

Data Source Use

Select a user-defined code (UDC) (98|SU) that indicates how the data source is configured. Select Servers (SVR) to run UBEs and business functions. Select Database (DB) to access table data.

Only database data sources are used to access data in tables.

Data Source Type

Select a UDC (98|DD) that indicates the type of database.

Data Class

Select a UDC (H96|DU) that describes the type of tables in the database, such as business data, control tables, and data dictionary.

Platform

Select a UDC (H93|PL) that indicates the type of physical hardware that the database resides on, such as **IBM Power Systems,Linux,** and **Sun Microsystems.**

Database Server Name

Select the enterprise server where the database or file resides.

Object Owner ID

Enter the database table prefix or owner. This field is available for these data source types:

- SQL Server OLEDB
- SQL Server ODBC
- SSE/ODBC.
- MSDE/ODBC
- MSDE/OLEDB
- OEE
- Oracle

Database Name

Enter the name that was assigned to the database during installation, such as **HPDEVORAP** or **HP9000.**

How this field is used depends on the data source type. If an Oracle data source is added, then this field contains the Oracle connect string. If a Microsoft Access, SQL*Server, or Client Access data source is added, then this field contains the Windows ODBC data source name. To minimize the number of connections to SQL*Server, the ODBC data source name should be the machine name where the SQL*Server database resides and the catalog name should be defined using the individual database name. For example, if you have two databases, DatabaseA and DatabaseB, on a machine called INTELNT, this field should contain INTELNT and the catalog name should be set to DatabaseA for one data source and DatabaseB for the other data source.

The data source name can be different from the ODBC or Oracle database name, if necessary.

Database Instance

Enter the name of the server instance. This name is required for these data source types:

- SQL Server OLEDB
- SQL Server ODBC
- SSE/ODBC.
- MSDE/ODBC
- MSDE/OLEDB

ODBC Data Source Name

Enter the name that is assigned to the database during installation, such as HPDEVORAP or HP9000. This field is available for these data source types:

- Access.
- SQL Server ODBC.
- DB2 UDB on OS/390.
- DB2 UDB on OS/400.
- SSE/ODBC.
- MSDE/ODBC.
- DB2 UDB on UNIX or Windows.

Schema

Enter the database table prefix or owner. This field is available for these data source types:

- DB2 UDB on OS/390.
- DB2 UDB on UNIX or Windows.

Database Alias Name

Enter the name of the database. This field is available for these data source types:

- DB2 UDB on OS/390.
- DB2 UDB on UNIX or Windows.

Library Name

Enter the name of the database. This field is available for the DB2 UDB on OS/400 data source type.

11.3 Adding Environments

The easiest way to add an environment to JD Edwards EnterpriseOne is to copy an existing environment.

This section discusses how to add Oracle's JD Edwards EnterpriseOne Environments.

11.3.1 Forms Used to Add Environments

Form Name	FormID	Navigation	Usage
Work With Environments	W0094E	EnterpriseOne Life Cycle Tools, System Administration Tools, System Administration Tools, Environment Management (GH9053), Environment Master	Select an environment to copy.
Copy Environment	W0094B	Click an environment to copy, and select Copy Environment from the Row menu on the Work With Environments form.	Enter a name for the new environment.

11.3.2 Adding Environments

Access the Work With Environments form.

1. Click the environment that most closely matches the environment that you want to create.

   For example, you might click the environment that you are signed in to or any other environment that you can access from the workstation.

2. From the Row menu, select Copy Environment.
3. On the Copy Environment form, enter a name for the new environment in the New Environment field.
4. To copy only the *PUBLIC OCM mappings of an environment, select Copy *PUBLIC Records Only and click OK.

   The Copy *PUBLIC Records Only option is selected by default. Clear the option to copy all mappings for the environment, including individual users and *PUBLIC.

See Also:

- "Setting Up Environments" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.

11.4 Setting Up Default OCM Mappings

This section provides an overview of OCM mappings and discusses how to set up default OCM mappings.

11.4.1 Understanding OCM Mappings

You must map all JD Edwards EnterpriseOne objects, including tables, by environment. You can map all objects of a specific object type using a single map, or you can define individual mappings by object name. When you define a default OCM mapping, you can select an existing environment and map the objects for that environment to the data sources where those objects exist.

You must create a default map for the table (TBLE) object type. Create this mapping with:

- Literal value of **DEFAULT** as the object name.

  This option enables you to map all tables in an environment to a specific data source.

- **TBLE** as the object type.

You must also indicate the data source where this table object resides. When you create a default map for the TBLE object type, all table objects point to the default data source unless the table has its own specific mapping. JD Edwards EnterpriseOne Table Conversion Design Aid uses this mapping for foreign tables.

To map a specific TBLE object type to a data source that is different from the DEFAULT, enter the table name as the object name. For example, F0002 is mapped to Control Tables-Test while the DEFAULT table mapping is mapped to Business Data-Test. The system reads the DEFAULT mapping for all tables except the F0002 table.

Each environment must have a default map for table objects for the *PUBLIC user profile because no inherent default location exists for table objects. If table objects do not have a default map and are not explicitly mapped by name, JD Edwards EnterpriseOne produces a Select/Failed error message when it tries to access the tables. Additionally, the tables do not appear in the input or output forms in JD Edwards EnterpriseOne Table Conversion Design Aid.

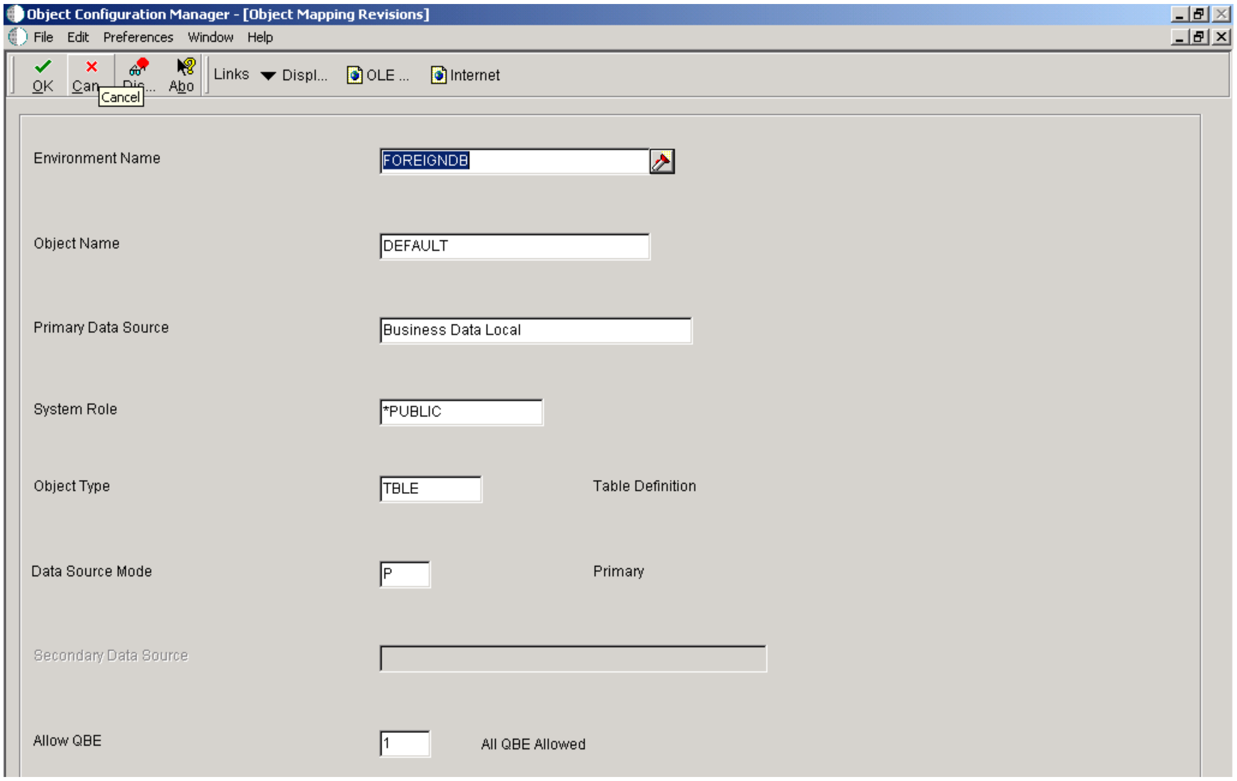
11.4.2 Forms Used to Set Up Default OCM Mappings

Form Name	FormID	Navigation	Usage
Machine Search and Select	W986110D	EnterpriseOne Life Cycle Tools, System Administration Tools, System Administration Tools (GH9011), Data Source Management, Object Configuration Manager	Select a data source.
Work With Object Mappings	W986110B	Select the appropriate machine on which the data source resides, and then click Select on the Machine Search & Select form.	Add, modify, and copy mappings.
Object Mapping Revisions	W986110C	Click Add on the Work With Object Mappings form.	Enter the environment name, object name, primary data source, system role, object type, and data source mode for a new mapping, and indicate whether to allow QBE (query by example) for the mapping.

11.4.3 Setting Up Default OCM Mappings

Access the Object Mapping Revisions form.

Figure 11–2 Object Mapping Revisions form



Object Configuration Manager - [Object Mapping Revisions]
File Edit Preferences Window Help
OK
Can
Cancel
Dis...
Abo
Links
Displ...
OLE ...
Internet

Environment Name	FOREIGNDB	
Object Name	DEFAULT	
Primary Data Source	Business Data Local	
System Role	*PUBLIC	
Object Type	TBLE	Table Definition
Data Source Mode	P	Primary
Secondary Data Source		
Allow QBE	1	All QBE Allowed

Environment Name

Select the name of the environment where the tables reside.

Object Name

Enter **DEFAULT** or the name of the specific object that you are mapping. The value **DEFAULT** creates a default map for all objects of a specific type.

Primary Data Source

Select the name of the primary data source.

System Role

Enter ***PUBLIC,** or select the JD Edwards EnterpriseOne user or role that had permissions to use these mappings. ***PUBLIC** gives all users permissions.

Object Type
Select a UDC (98 | OM) that indicates the type of object for which you are creating a mapping.

Data Source Mode
Select a UDC (98 | DM) that indicates whether the primary or secondary data source should be used.

Secondary Data Source
Select the name of the secondary data source. This field is available only when **secondary** has been selected for the data source mode. This data source is used if the primary data source or the data item in the primary data source cannot be located.

Allow QBE (query by example)
Select a UDC (98 | QB) that indicates whether row-level record locking is selected for the data source. Row-level record locking should be selected to help prevent database integrity issues.

See Also:

- "Understanding Object Configuration Manager" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.
- "Understanding Data Sources" in the *JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation Guide*.

Glossary

Accessor Methods/Assessors

Java methods to “get” and “set” the elements of a value object or other source file.

activity rule

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

add mode

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

Advanced Planning Agent (APAg)

A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of rational databases, flat file format, and other data or message encoding, such as XML.

application server

Software that provides the business logic for an application program in a distributed environment. The servers can be Oracle Application Server (OAS) or WebSphere Application Server (WAS).

Auto Commit Transaction

A database connection through which all database operations are immediately written to the database.

batch processing

A process of transferring records from a third-party system to JD Edwards EnterpriseOne.

In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.

batch server

A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.

batch-of-one

A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks.

best practices

Non-mandatory guidelines that help the developer make better design decisions.

BPEL

Abbreviation for Business Process Execution Language, a standard web services orchestration language, which enables you to assemble discrete services into an end-to-end process flow.

BPEL PM

Abbreviation for Business Process Execution Language Process Manager, a comprehensive infrastructure for creating, deploying, and managing BPEL business processes.

Build Configuration File

Configurable settings in a text file that are used by a build program to generate ANT scripts. ANT is a software tool used for automating build processes. These scripts build published business services.

build engineer

An actor that is responsible for building, mastering, and packaging artifacts. Some build engineers are responsible for building application artifacts, and some are responsible for building foundation artifacts.

Build Program

A WIN32 executable that reads build configuration files and generates an ANT script for building published business services.

business analyst

An actor that determines if and why an EnterpriseOne business service needs to be developed.

business function

A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules, and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.

business function event rule

See named event rule (NER).

business service

EnterpriseOne business logic written in Java. A business service is a collection of one or more artifacts. Unless specified otherwise, a business service implies both a published business service and business service.

business service artifacts

Source files, descriptors, and so on that are managed for business service development and are needed for the business service build process.

business service class method

A method that accesses resources provided by the business service framework.

business service configuration files

Configuration files include, but are not limited to, interop.ini, JDBj.ini, and jdelog.properties.

business service cross reference

A key and value data pair used during orchestration. Collectively refers to both the code and the key cross reference in the WSG/XPI based system.

business service cross-reference utilities

Utility services installed in a BPEL/ESB environment that are used to access JD Edwards EnterpriseOne orchestration cross-reference data.

business service development environment

A framework needed by an integration developer to develop and manage business services.

business services development tool

Otherwise known as JDeveloper.

business service EnterpriseOne object

A collection of artifacts managed by EnterpriseOne LCM tools. Named and represented within EnterpriseOne LCM similarly to other EnterpriseOne objects like tables, views, forms, and so on.

business service framework

Parts of the business service foundation that are specifically for supporting business service development.

business service payload

An object that is passed between an enterprise server and a business services server. The business service payload contains the input to the business service when passed to the business services server. The business service payload contains the results from the business service when passed to the Enterprise Server. In the case of notifications, the return business service payload contains the acknowledgement.

business service property

Key value data pairs used to control the behavior or functionality of business services.

Business Service Property Admin Tool

An EnterpriseOne application for developers and administrators to manage business service property records.

business service property business service group

A classification for business service property at the business service level. This is generally a business service name. A business service level contains one or more business service property groups. Each business service property group may contain zero or more business service property records.

business service property key

A unique name that identifies the business service property globally in the system.

business service property utilities

A utility API used in business service development to access EnterpriseOne business service property data.

business service property value

A value for a business service property.

business service repository

A source management system, for example ClearCase, where business service artifacts and build files are stored. Or, a physical directory in network.

business services server

The physical machine where the business services are located. Business services are run on an application server instance.

business services source file or business service class

One type of business service artifact. A text file with the .java file type written to be compiled by a Java compiler.

business service value object template

The structural representation of a business service value object used in a C-business function.

Business Service Value Object Template Utility

A utility used to create a business service value object template from a business service value object.

business services server artifact

The object to be deployed to the business services server.

business view

A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.

central objects merge

A process that blends a customer's modifications to the objects in a current release with objects in a new release.

central server

A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.

charts

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

check-in repository

A repository for developers to check in and check out business service artifacts. There are multiple check-in repositories. Each can be used for a different purpose (for example, development, production, testing, and so on).

checksum

A fixed-size datum computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage. JD Edwards EnterpriseOne uses the checksum to verify the integrity of packages that have been downloaded by recomputing the checksum of the downloaded package and comparing it with the checksum of the original package. The procedure that yields the checksum from the data is called a checksum function or checksum algorithm. JD Edwards EnterpriseOne uses the MD5 and STA-1 checksum algorithms.

connector

Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.

Control Table Workbench

An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.

control tables merge

A process that blends a customer's modifications to the control tables with the data that accompanies a new release.

correlation data

The data used to tie HTTP responses with requests that consist of business service name and method.

credentials

A valid set of JD Edwards EnterpriseOne username/password/environment/role, EnterpriseOne session, or EnterpriseOne token.

cross-reference utility services

Utility services installed in a BPEL/ESB environment that access EnterpriseOne cross-reference data.

database credentials

A valid database username/password.

database server

A server in a local area network that maintains a database and performs searches for client computers.

Data Source Workbench

An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

deployment artifacts

Artifacts that are needed for the deployment process, such as servers, ports, and such.

deployment server

A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

direct connect

A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one and store-and-forward.

Do Not Translate (DNT)

A type of data source that must exist on the iSeries because of BLOB restrictions.

embedded application server instance

An OC4J instance started by and running wholly within JDeveloper.

edit code

A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

edit mode

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

edit rule

A method used for formatting and validating user entries against a predefined rule or set of rules.

Electronic Data Interchange (EDI)

An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

embedded event rule

An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.

Employee Work Center

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

enterprise server

A server that contains the database and the logic for JD Edwards EnterpriseOne.

Enterprise Service Bus (ESB)

Middleware infrastructure products or technologies based on web services standards that enable a service-oriented architecture using an event-driven and XML-based messaging framework (the bus).

EnterpriseOne administrator

An actor responsible for the EnterpriseOne administration system.

EnterpriseOne credentials

A user ID, password, environment, and role used to validate a user of EnterpriseOne.

EnterpriseOne development client

Historically called “fat client,” a collection of installed EnterpriseOne components required to develop EnterpriseOne artifacts, including the Microsoft Windows client and design tools.

EnterpriseOne extension

A JDeveloper component (plug-in) specific to EnterpriseOne. A JDeveloper wizard

is a specific example of an extension.

EnterpriseOne object

A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

EnterpriseOne process

A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

EnterpriseOne resource

Any EnterpriseOne table, metadata, business function, dictionary information, or other information restricted to authorized users.

Environment Workbench

An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

escalation monitor

A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

event rule

A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

explicit transaction

Transaction used by a business service developer to explicitly control the type (auto or manual) and the scope of transaction boundaries within a business service.

exposed method or value object

Published business service source files or parts of published business service source files that are part of the published interface. These are part of the contract with the customer.

fast path

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

file server

A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

final mode

The report processing mode of a processing mode of a program that updates or creates data records.

foundation

A framework that must be accessible for execution of business services at runtime. This includes, but is not limited to, the Java Connector and JDBj.

FTP server

A server that responds to requests for files via file transfer protocol.

HTTP Adapter

A generic set of services that are used to do the basic HTTP operations, such as GET, POST, PUT, DELETE, TRACE, HEAD, and OPTIONS with the provided URL.

instantiate

A Java term meaning “to create.” When a class is instantiated, a new instance

is created.

integration developer

The user of the system who develops, runs, and debugs the EnterpriseOne business services. The integration developer uses the EnterpriseOne business services to develop these components.

integration point (IP)

The business logic in previous implementations of EnterpriseOne that exposes a document level interface. This type of logic used to be called XBPs. In EnterpriseOne 8.11, IPs are implemented in Web Services Gateway powered by webMethods.

integration server

A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

integrity test

A process used to supplement a company’s internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

interface table

See Z table.

internal method or value object

Business service source files or parts of business service source files that are not part of the published interface. These could be private or protected methods. These could be value objects not used in published methods.

interoperability model

A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

in-your-face error

In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

jargon

An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

Java application server

A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

JDBNET

A database driver that enables heterogeneous servers to access each other's data.

JDEBASE Database Middleware

A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

JDECallObject

An API used by business functions to invoke other business functions.

jde.ini

A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

JDEIPC

Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

jde.log

The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

JDENET

A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

JDeveloper Project

An artifact that JDeveloper uses to categorize and compile source files.

JDeveloper Workspace

An artifact that JDeveloper uses to organize project files. It contains one or more project files.

JMS Queue

A Java Messaging service queue used for point-to-point messaging.

listener service

A listener that listens for XML messages over HTTP.

local repository

A developer's local development environment that is used to store business service artifacts.

Location Workbench

An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

logic server

A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

MailMerge Workbench

An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

Manual Commit transaction

A database connection where all database operations delay writing to the database until a call to commit is made.

master business function (MBF)

An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

master table

See published table.

media storage object

Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

message center

A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

messaging adapter

An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

messaging server

A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

Monitoring Application

An EnterpriseOne tool provided for an administrator to get statistical information for various EnterpriseOne servers, reset statistics, and set notifications.

named event rule (NER)

Encapsulated, reusable business logic created using event rules, rather that C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

Object Configuration Manager (OCM)

In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

Object Librarian

A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-incapabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as production and development) and enables objects to be easily moved from one environment to another.

Object Librarian merge

A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

Open Data Access (ODA)

An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

Output Stream Access (OSA)

An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

package

JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

package build

A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: “Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions.” The process of creating a package build is often referred to, as it is in this example, simply as “a package build.”

package location

The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

Package Workbench

An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

Pathcode Directory

The specific portion of the file system on the EnterpriseOne development client where EnterpriseOne development artifacts are stored.

patterns

General repeatable solutions to a commonly occurring problem in software design. For business service development, the focus is on the object relationships and interactions.

For orchestrations, the focus is on the integration patterns (for example, synchronous and asynchronous request/response, publish, notify, and receive/reply).

print server

The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

pristine environment

A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

processing option

A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

production environment

A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

Production Published Business Services Web Service

Published business services web service deployed to a production application server.

program temporary fix (PTF)

A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

project

In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

promotion path

The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, 11 equals new project pending review, 21 equals programming, 26 equals QA test/review, 28 equals QA test/review complete, 38 equals in production, 01 equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

proxy server

A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

published business service

EnterpriseOne service level logic and interface. A classification of a published business service indicating the intention to be exposed to external (non-EnterpriseOne) systems.

published business service identification information

Information about a published business service used to determine relevant authorization records. Published business services + method name, published business services, or *ALL.

published business service web service

Published business services components packaged as J2EE Web Service (namely, a J2EE EAR file that contains business service classes, business service foundation, configuration files, and web service artifacts).

published table

Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

publisher

The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

QBE

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

real-time event

A message triggered from EnterpriseOne application logic that is intended for external systems to consume.

refresh

A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level.

replication server

A server that is responsible for replicating central objects to client machines.

rules

Mandatory guidelines that are not enforced by tooling, but must be followed in order to accomplish the desired results and to meet specified standards.

secure by default

A security model that assumes that a user does not have permission to execute an object unless there is a specific record indicating such permissions.

Secure Socket Layer (SSL)

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

selection

Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.

serialize

The process of converting an object or data into a format for storage or transmission across a network connection link with the ability to reconstruct the original data or objects when needed.

Server Workbench

An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number data source. The application also updates the Server Plan detail record to reflect completion.

SOA

Abbreviation for Service Oriented Architecture.

softcoding

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

source repository

A repository for HTTP adapter and listener service development environment artifacts.

Specification merge

A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.

specification

A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.

Specification Table Merge Workbench

An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.

SSL Certificate

A special message signed by a certificate authority that contains the name of a user and that user's public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in the user's public key.

store-and-forward

The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.

subscriber table

Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.

super class

An inheritance concept of the Java language where a class is an instance of something, but is also more specific. “Tree” might be the super class of “Oak” and “Elm,” for example.

table access management (TAM)

The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.

Table Conversion Workbench

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

table conversion

An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.

table event rules

Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.

terminal server

A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

transaction processing (TP) monitor

A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

transaction processing method

A method related to the management of a manual commit transaction boundary (for example, start, commit, rollback, and cancel).

transaction set

An electronic business transaction (electronic data interchange standard document) made up of segments.

trigger

One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

triggering event

A specific workflow event that requires special action or has defined consequences or resulting actions.

user identification information

User ID, role, or *public.

User Overrides merge

Adds new user override records into a customer’s user override table.

value object

A specific type of source file that holds input or output data, much like a data structure passes data. Value objects can be exposed (used in a published business service) or internal, and input or output. They are comprised of simple and complex elements and accessories to those elements.

versioning a published business service

Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

Versions List merge

The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

visual assist

Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

vocabulary override

An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

web application server

A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

web server

A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

Web Service Description Language (WSDL)

An XML format for describing network services.

Web Service Inspection Language (WSIL)

An XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made.

web service softcoding record

An XML document that contains values that are used to configure a web service proxy. This document identifies the endpoint and conditionally includes security information.

web service softcoding template

An XML document that provides the structure for a soft coded record.

Where clause

The portion of a database operation that specifies which records the database operation will affect.

Windows terminal server

A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

wizard

A type of JDeveloper extension used to walk the user through a series of steps.

workbench

A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

workflow

The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

workgroup server

A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

XAPI events

A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

XML CallObject

An interoperability capability that enables you to call business functions.

XML Dispatch

An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

XML List

An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

XML Service

An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

XML Transaction

An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

XML Transaction Service (XTS)

Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

Z event

A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

Z table

A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

Z transaction

Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.

Index