



# JD Edwards EnterpriseOne Tools

---

## Development Tools Overview Guide

**Release 9.2**



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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Preface

Welcome to the JD Edwards EnterpriseOne Tools Development Tools: Overview Guide.

## Audience

This guide is intended for system administrators and technical consultants.

## Documentation Accessibility

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

## Access to Oracle Support

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

## Related Information

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

<http://learnjde.com>

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. See the [JD Edwards EnterpriseOne Tools Server Manager Guide](#).

## Conventions

The following text conventions are used in this document:

| Convention             | Meaning   |
|------------------------|---|
| <b>boldface</b>        | Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.     |
| <i>italics</i>         | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.                     |
| <code>monospace</code> | Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter. |



# 1 Understanding JD Edwards EnterpriseOne Acronyms

## Acronym List

Many of the acronyms that are commonly used in EnterpriseOne Development Tools are defined in the following table:

| Term       | Description                     |
|------------|---------------------------------|
| BDA        | Business View Design Aid        |
| BSFN       | Business Function               |
| BSVW       | Business View                   |
| CSV        | Comma Separated Values          |
| DD         | Data Dictionary                 |
| DS or DSTR | Data Structure                  |
| ER         | Event Rules                     |
| FDA        | Form Design Aid                 |
| H4A        | HTML for Applications           |
| NER        | Named Event Rule                |
| OCM        | Object Configuration Manager    |
| OL         | Object Librarian                |
| OMC        | Object Management Configuration |
| OMW        | Object Management Workbench     |
| OSA        | Output Stream Access            |
| PO         | Processing Option               |

| Term | Description              |
|------|--------------------------|
|      |                          |
| QBE  | Query by Example         |
| RDA  | Report Design Aid        |
| TAM  | Table Access Management  |
| TBLE | Table                    |
| TC   | Table Conversions        |
| TDA  | Table Design Aid         |
| TER  | Table Event Rule         |
| UBE  | Universal Batch Engine   |
| UDC  | User Defined Code        |
| UTB  | Universal Table Browser  |
| WF   | Workflow                 |
| XREF | Cross Reference Facility |

# 2 Understanding JD Edwards EnterpriseOne Development Tools

## Understanding JD Edwards EnterpriseOne Development Tools

**Note:** The following topics will be discussed in more detail later in the document.

### JD Edwards EnterpriseOne Development Tools

The development tools that JD Edwards EnterpriseOne Tools provides offer a powerful application development environment in which you can build and customize EnterpriseOne applications to suit your specific needs quickly and easily. By using EnterpriseOne Tools development tools you can complete a variety of tasks including:

- Design and define application objects.
- Enable applications to serve different locations and languages while sharing the same data.
- Define end-to-end processes in a user-friendly, graphical design environment.

### Object Management Workbench

Object Management Workbench (OMW) manages all EnterpriseOne objects. Developers use OMW to create new objects and check out existing objects from a central development environment, copying those objects to their workstation. They can then use the development tools to change objects and check them back in for others to access.

### Data Dictionary

Just as a dictionary contains word definitions, the data dictionary is a central repository that contains data item definitions and attributes. These attributes determine how a data item:

- Appears on reports and forms.
- Validates data entry within an application.
- Assigns column and row descriptions.
- Provides text for field help

- Is stored in a table.

## Table Design

A relational database table is used to store the data that an application uses. Although a new application might use one or more tables that already exist, you can use the Table Design Aid to create new tables if the application requires it. To create a table, you select data items and then assign key fields as indexes for retrieving and updating data.

## Business View Design

Business views are the link between applications and data. A business view defines the data items from one or more tables that an application uses. After you determine the data items needed by an application, you can create a new business view if you are not able to use an existing one. With business views, you can select only the data items needed in the application, which increases performance due to less data moving over the network. For example, you could create a business view that contains only employee names and addresses from a table containing all employee data.

## Form Design

Use Form Design Aid (FDA) to create one or more forms for an application. A form is a graphical user interface that enables users to interact with the system. A form can be used to search and display data, as well as enter new data and modify existing data. A single application can contain one or more forms. To create an application, determine the type of form the application requires and associate each form with a business view. To design forms, you add controls such as a grid, edit fields, push buttons, and radio buttons.

Usually, a find/browse form is the first form that appears in the application. It enables the user to locate a specific record with which to work. Upon selecting a record, a subsequent form such as a fix/inspect form can be used to provide details of the record. Power forms enable you to design applications that use one single power form to locate a specific record and display its detail records on one form.

## Data Structure Design

Data structures are composed of data items defined in the data dictionary and are used to pass data to and from interactive and batch applications. You use Data Structure Design to create and modify EnterpriseOne data structures.

## Event Rules

Events are activities that occur on a form, such as when a user enters information into a field or exits a field by using the Tab key. Events can be initiated by the user or by the application. Event rules (ER) are logic statements that you can



create and attach to events. ER is initiated when events occur at runtime. You can attach multiple event rules to one event. The various kinds of event rules include:

- Conditional statements, such as If/Else/End If.
- While loops.
- Assignments.
- Calls to business functions.
- Form or report interconnections.
- Calls to system functions.
- Table I/O operations.

## System Functions

System functions are predefined sets of logic shipped with the JD Edwards EnterpriseOne product. These functions enable you to perform specialized processing without adding custom code. You use system functions within JD Edwards EnterpriseOne Report Design Aid (RDA), Form Design Aid (FDA), and Workflow. Each of these has a specific set of system functions that apply to it.

## Business Functions

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

You can use master business functions to update master files (such as Address Book Master and Item Master) or to update transaction files (such as sales orders and purchase orders). Generally, master file master business functions, which access tables, are simpler than transaction file master business functions, which are specific to a program. Transaction master business functions provide a common set of functions that contain all of the necessary default values and editing for a transaction file. Transaction master business functions contain logic that ensures the integrity of the transaction being inserted, updated, or deleted from the database.

## APIs

APIs are routines that perform predefined tasks. JD Edwards EnterpriseOne APIs make it easier for third-party applications to interact with JD Edwards EnterpriseOne software. These APIs are functions that you can use to manipulate data types, provide common functionality, and access the database. Several categories of APIs exist, including the Common Library Routines and JD Edwards EnterpriseOne Database (JDEBASE) APIs. Programming with APIs is useful for these reasons:

- No code modifications are required as functionality is upgraded.
- When a data structure changes, source modifications are minimal to nonexistent.
- Common functionality is provided through the APIs, and they are less prone to error.

- When the code in an API changes, business functions typically only need to be recompiled and relinked.

## Report Design

You can use Report Design Aid (RDA) to create a variety of simple and complex batch processes and reports. The interface is simple enough to use without programming expertise, yet powerful enough to create the most complex reports. You can also use RDA to create batch processes and reports. Report Design Aid includes a director to guide you through the process of creating report templates. This Report Director presents multiple report components from which to choose. You can create custom directors to aid in the creation of report templates. These directors are configured to use report components to meet a specific reporting requirement. After using the director to create the initial report, you can enhance the report by:

- Inserting additional report sections
- Modifying properties
- Adding logic
- Further organizing the data
- Calculating totals

The design work space in RDA can be configured to accommodate individual work preferences. You can:

- Modify the report view options.
- Select which toolbars and windows to display.
- Arrange windows.

You can use RDA with terminal server. Just like in a traditional client server configuration, a report template that is checked out using terminal server cannot be accessed by other users.

## Batch Versions

Batch Versions is a tool that you use to create and process versions of report templates. You can use Batch Versions to:

- Add and copy batch versions.
- Define processing options, data selection and data sequencing.
- Check out batch versions, check in batch versions, erase the check out, and copy version specifications to the enterprise server.
- Access RDA to modify batch versions without changing the report template specifications.
- Submit batch versions for processing and override processing options, data selection, and data sequencing at runtime.
- Review batch version processing by using BrowsER, the report cover page, and logs for reporting.

## Workflow

EnterpriseOne Workflow Tools enables you to automate a high-volume, formerly paper-based process into an email-based process flow across a network. Documents, information, and tasks pass from one participant to another

for action based on a set of procedural rules. The result is an automated and efficient process with minimal user involvement, which enables you to streamline existing business processes, increase efficiency, and reduce process time.



# 3 Understanding Objects

## Understanding Objects

In JD Edwards EnterpriseOne, an object is a reusable entity that is based on software specifications created by the JD Edwards EnterpriseOne Tools.

A specification is a complete description of a JD Edwards EnterpriseOne object. Specifications can be thought of as metadata. Each object has its own specification, which is stored on both the server and the workstation. Some specifications describe types of objects; for example, data structure specifications can describe business function data structures, processing option structures, or media object structures.

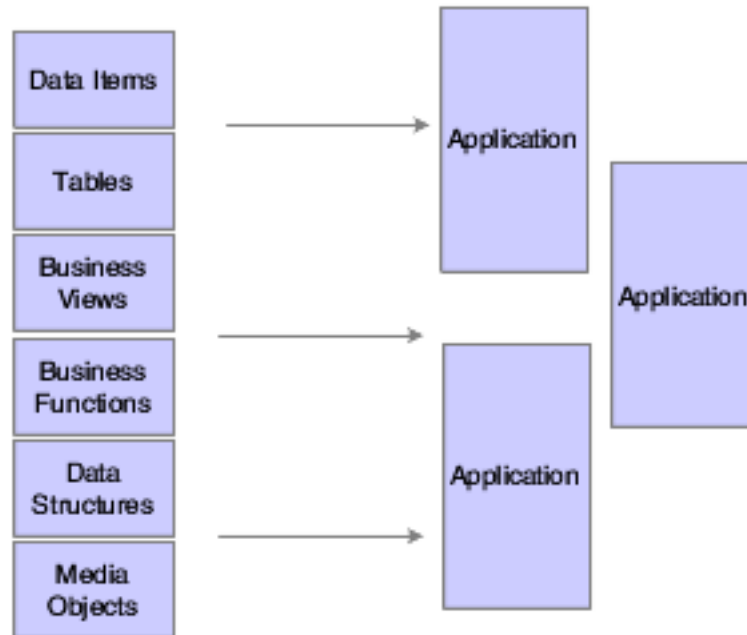
JD Edwards EnterpriseOne architecture is object-based, which means that discrete software objects are the basis for all applications and that developers can reuse the objects in multiple applications. This use of objects (applications being broken down into smaller components) allows JD Edwards EnterpriseOne to provide true distributed processing. Developers create objects using JD Edwards EnterpriseOne Tools.

See *"Working with Objects" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

Examples of JD Edwards EnterpriseOne objects include the following:

- Batch applications
- Business functions (encapsulated routines)
- Business views
- Data dictionary items
- Data structures
- Event rules
- Interactive applications
- Media objects
- Tables

This diagram shows components of the object model:



## Understanding How JD Edwards EnterpriseOne Stores Objects

JD Edwards EnterpriseOne stores objects in the following two places:

- A central-storage server stores central objects. Central objects reside in a central location from which you can deploy them. Other objects, such as specifications, are stored in a relational database. Still others, such as DLLs and source code, are stored on a file server.
- Any machine (workstation or server) that runs JD Edwards EnterpriseOne stores replicated objects. A copy (replicated) set of the central objects must reside on each development workstation and server that runs JD Edwards EnterpriseOne. The path code indicates the directory in which these objects are located.

To move objects between the server and workstation, you use the check-in and check-out options in JD Edwards Object Management Workbench. When you create an object, it initially resides on your workstation. Unless you check it into the server, it is available only to you. After you check it into the server, it is available for other users to check out. When you check out an object, all object specification records (a collection of data that defines an EnterpriseOne object) are copied from the server to your workstation.

See [JD Edwards EnterpriseOne Tools Object Management Workbench Guide](#) .

# 4 Understanding Creating and Managing Objects

## Understanding Object Management Workbench

Object Management Workbench is the primary component of the change management system for JD Edwards EnterpriseOne development. A change management system is vital to a productive development environment because it helps organize a myriad of development activities and helps prevent problems, such as when a developer intermixes components from different releases or when multiple developers simultaneously change an object. Object Management Workbench automates many of these change management activities.

This section discusses:

- OMW projects
- Allowed actions
- Tokens
- The OMW interface
- Object Librarian and non-Object Librarian objects

## OMW Projects

Projects are composed of objects and owners. All development of objects within JD Edwards EnterpriseOne must be performed within the context of a project. Usually, you must first create or select a project, add an object to it, and then you can work with that object. Typically, objects are included in a project because they have been modified or created by a developer to complete a specific task.

In addition to objects, users can be associated with different projects. In fact, before you can add an object to a project, you must have been added to the project as a user in a role that has permission to add objects. A user can be assigned to the same project more than once with different roles. Projects may also contain other projects.

See *"Working with Projects" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

## Allowed Actions

Allowed actions are rules that define the actions that may be performed by a user who is assigned a specific user role. You set up these rules for each user role, object type, and project status by using the Object Management Workbench Configuration program.

See *"Configuring User Roles and Allowed Actions" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide*

## Tokens

Some objects use tokens to minimize the possibility of one user overwriting another user's changes to an object. The token management system organizes application development by providing a single checkout environment. Tokens provide a change control solution in a system that does not support merging or multiple versions of object specifications.

**Note:** Only Object Librarian objects have tokens.

See *"Working with Tokens" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

## The OMW Interface

From left to right, the initial OMW form displays these features:

- The project window, which displays your projects and their related objects and users. To view your current projects, click Find.
- The center column, which contains action buttons that you use to perform actions on a selected object. Available buttons vary based on your roles in the current project and on the status of the project in which the selected object resides. When you first launch OMW, no buttons appear in the center column because you have not selected an object.
- The information window, which displays a Web site; project status and release information; object or user information; and search results. Initially, the window displays a Web site or HTML page. The contents change based on your tab and object selections. For example, when you select a project or an object in the project window, the information window displays information about the selected project or object. To return this window to its initial state, click News on the toolbar.

See *"The JD Edwards EnterpriseOne OMW Interface" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

## Object Librarian and Non-Object Librarian Objects

OMW provides control of EnterpriseOne objects in a simple, integrated, graphical user interface for software development. In EnterpriseOne, an object is a reusable entity based on software specifications that are created by the EnterpriseOne development tools.

In OMW, this definition is expanded to include non-Object Librarian objects that are data source-based rather than path code-based.

JD Edwards EnterpriseOne objects include the following Object Librarian objects:

- Batch applications and versions
- Business functions



- Business views
- Data structures
- Interactive applications
- Media objects
- Tables

EnterpriseOne objects include the following non-Object Librarian objects:

- Data dictionary items
- User defined code items
- Workflow objects

See *"Object Librarian and Non-Object Librarian Objects" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .



# 5 Creating and Maintaining Applications

## Understanding Applications

An application is a collection of objects that performs a specific task. You use JD Edwards EnterpriseOne Tools to build standard groups of related applications, such as: Architecture, engineering, and construction; Distribution; Energy and chemical systems; Financial applications; Workforce management; Manufacturing; and Technical applications. These applications share a common user interface because they are all generated through JD Edwards EnterpriseOne Tools. Applications refer to both interactive and batch applications. For example, all of the following are applications:

- Address Book Revisions
- Sales Order Entry
- General Ledger Post
- Trial Balance Report

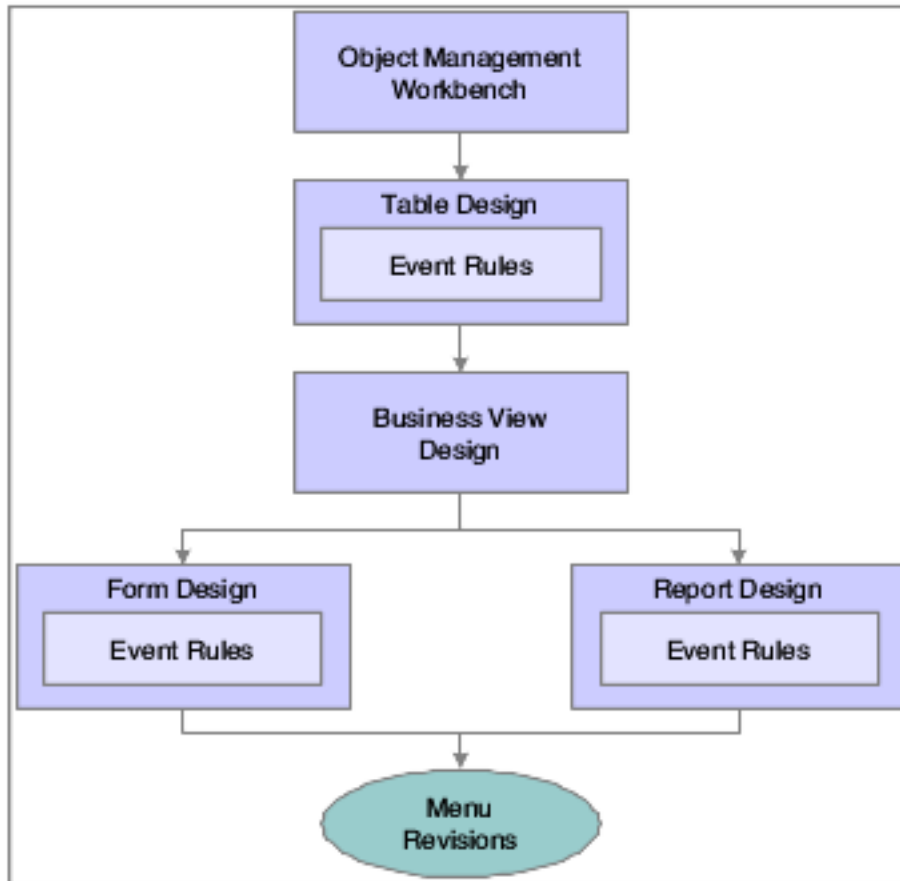
See *JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .

## Creating Applications

You use JD Edwards EnterpriseOne Tools to build your applications. You always begin your application development from the JD Edwards EnterpriseOne Object Management Workbench. You might not use every tool to create an application. For example, if you don't need to add or modify data items you would proceed to Table Design from the Object Management Workbench. If one or more existing database tables contain all of the data items that you want to include in your application, you can skip the step of designing a table and proceed to Business View Design.

This flowchart illustrates the Development Cycle:

See *"Interactive Application Fundamentals" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .



## Understanding Data Items and the Data Dictionary

A data item identifies a unit of information. The data item definition defines how the item can be used and includes information such as the type of item and its length.

Because the data dictionary is dynamic, any changes that you make to a data item are effective immediately for all applications that include the data item. Applications access the data dictionary at runtime and immediately reflect modifications to data item attributes such as field descriptions, column headings, decimals, and edit rules.

You use the data dictionary to create, view, and update attributes for data items. You can copy a data item with similar attributes and modify it for your specific needs. This method can be quicker and easier than creating a new data item, but if you use this method you must distinguish between the original and the copy. You distinguish between them by modifying the alias.

Because changes to a data item are immediately reflected throughout the JD Edwards EnterpriseOne tools at runtime, remember that changing the type and attributes of a data item might affect how your data is stored and cause discrepancies among records.

See *"Understanding the Data Dictionary" in the JD Edwards EnterpriseOne Tools Data Dictionary Guide* .

## Understanding Table Design

A relational database table stores the data that an application uses in columns and rows. Each column is a data item, and each row is a record. You can create one or more tables for use in an application. To create a table, you select data items (the data items must already exist in the data dictionary) to include in the table and assign key fields as indices for retrieving and updating data. You must define your table so that JD Edwards EnterpriseOne software recognizes that the table exists.

You must use Table Design to generate the table whenever you want to:

- Create a new table.
- Add or delete a data item.
- Add or modify an index.

An index identifies records in a table. A primary index identifies unique records in a table. An index is composed of one or more keys, or data items, within the table. An index enables a database management system (DBMS) to sort and locate records quickly.

See *"Understanding JD Edwards EnterpriseOne Table Design Aid" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .

## Understanding Business View Design

A business view is a selection of data items from one or more tables. After you create a table, use Business View Design to choose only the data items that are required for your application. EnterpriseOne uses the business view that you define to generate the appropriate SQL statements necessary to retrieve data from any of the supported databases. After you define a business view, you can create a form that updates data in an interactive application or you can design a report that displays data. Because you choose only those data items that an application requires, less data moves over the network.

Business views are required for creating applications and generating reports; they have the following characteristics:

- Contain some or all of the data items from one or more tables.
- Link a JD Edwards EnterpriseOne application to one or more tables.
- Define the data items from multiple tables used by an application (such as table joins or table unions).

See *"Understanding Business Views" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .

## Understanding Form Design Aid

Form Design Aid is part of the Interactive Application Design and is used to create or modify EnterpriseOne applications. Applications are composed of forms, and a form is the interface between a user and a table. This interface should present the data logically and contain the functions that are necessary to enter and manipulate data.

Interactive Application Design is the entry point to several tools for creating, generating, running, maintaining, and securing applications. Interactive Application Design includes Form Design Aid for creating forms and Event Rules Design for attaching business logic through event rules. Use Interactive Application Design to do the following:

- Access Form Design for creating forms.
- Run an application
- Create text overrides
- Browse ER
- Browse forms in an application
- Use Visual ER Compare to compare event rules between two versions of an application
- Use FDA Compare to compare one version of an application with another

To start Interactive Application Design, choose an application in Object Management Workbench and click the Design button. In Interactive Application Design, you can change the metadata for the application. To access application metadata, click the Summary, Category Codes, and Install/Merge Codes tabs. You can also attach text and files to an application by clicking the Attachments tab. You can access all other functions from the Design Tools tab.

See *JD Edwards EnterpriseOne Tools Form Design Aid Guide* .

## Understanding Report Design Aid

Report Design Aid is used to present business data stored in the EnterpriseOne database. EnterpriseOne data is stored in databases using relational tables. The data is typically presented using batch applications that access the data through business views.

You can use Report Design Aid to create a variety of simple and complex batch processes and reports. The interface is simple enough to use without programming expertise, yet powerful enough to create the most complex reports. You can also use Report Design Aid to create batch processes and reports.

Each report is comprised of sections, which are the building blocks of all reports. Within the template, you can add, hide, remove, and rearrange sections as needed.

You cannot process a report without a batch version. The batch version is submitted for processing, and once submitted, runs without user interaction. You do not interact with the report again until processing is complete.

A report exists as a set of specifications that are read by the EnterpriseOne batch engine for processing. You can create variations of a single report template using batch versions. The first step in creating a report is to create a report object within EnterpriseOne. This report is actually a template from which multiple versions can be created.

See *"Creating Reports" in the JD Edwards EnterpriseOne Tools Report Design Aid Guide* .

## Understanding Data Structure Design

Data structures are a key element of any programming language or environment. A data structure is a list of parameters that passes data among applications and tables or forms. JD Edwards EnterpriseOne uses data structures in the following instances:

- The system generates a data structure.

- You create a data structure.

The two types of system-generated data structures are as follows:

- **Form**  
Each form with an attached business view has a default data structure. Data structures receive parameters from or send parameters to other forms during Form Interconnects. You maintain the data structure by using the Form/Data Structure menu option in Form Design.
- **Report**  
A batch application with an attached business view can receive parameters from or send parameters to a data structure. You can create and maintain the data structure from the Report/Data Structure menu option in Report Design. Unlike a form data structure, this type of data structure is not automatically populated with data items.

As a user, you can create three types of data structures, as follows:

- **Media object data structures**  
To enable an application for media objects, you must create a data structure to pass arguments from the application table to the media object table. To work with a data structure for media objects, create a new media object data structure or select an existing one to modify in Object Management Workbench.
- **Processing options data structures**  
You use processing options to create an input property sheet. You use a parameter list to pass processing options to an application. You can create a processing option data structure template or modify an existing template in Object Management Workbench.
- **Business function data structures**  
Any business function, whether it uses C or Business Function Event Rules as its source language, must have a defined data structure to send or receive parameters to or from applications. You can create a DSTR object type, or choose an existing object type to work with in Object Management Workbench. You can also create data structures for text substitution messages. Additionally, you can attach notes, such as an explanation of use, to any data structure or data item within the structure.

See [JD Edwards EnterpriseOne Tools Data Structure Design Guide](#) .

## Understanding Event Rules Design

Use Event Rules Design to create business logic for an application. You can create event rules that do the following:

- Perform a mathematical calculation.
- Pass data from a field on a form to a field on another form.
- Count grid rows that are populated with data.
- Interconnect two forms.
- Hide or display a control using a system function.
- Evaluate If/While and Else conditions.
- Assign a value or an expression to a field.
- Create variables or programmer-defined fields at runtime.

- Perform a batch process upon completion of an interactive application.
- Process table input and output, validate data, and retrieve records.

Areas where event rules can be added are:

- Controls

A control is a reusable object that appears on a form. Examples include push buttons, edit fields, and grids. A form itself is also considered a control. Controls can be simple or complex. Simple controls have few event points to which logic can be attached. Complex controls can have many event points to which logic can be attached

- Events

Events are activities that occur on a form, such as entering information a form or exiting a field by using the Tab key. Events can be initiated by the user or the application. A single control might initiate multiple events. The system also initiates some events, such as Last Grid Record Read, when certain actions occur

- Form Processing

Form processing refers to the business logic associated with each form. By default, each type of EnterpriseOne form automatically processes various events. You specify additional logic by using Event Rules Design. Form processing depends on the occurrence of specific events, such as initializing a form or changing the value of a field

- Event Rules

Event rules are logic statements that you can create and attach to events. JD Edwards EnterpriseOne software uses two types of event rules: business function event rules and embedded event rules. Event rules are initiated when events occur at runtime. You can attach multiple event rules to one event. The various kinds of event rules include:

- Business Function Event Rules

Business function event rules are encapsulated, reusable, business logic that you create using Event Rules Design, rather than C programming. Business function event rules are stored as objects and are compiled. Business function event rules are sometimes called Named Event Rules (NERs).

- Embedded Event Rules

Embedded event rules are specific to a particular table, interactive application, or batch application. They are not reusable. Examples include using form-to-form calls, hiding a field that is based on a value in a processing option, and calling a business function. Embedded event rules can be in application event rules (interactive or batch) or in table event rules. They can be compiled or uncompiled.

- Application Event Rules

You can add business logic that is specific to a particular application. Interactive applications connect event rules via Form Design, while batch event rules use Report Design.

- Table Event Rules

You can create database triggers, or rules that you attach to a table by using Table Design Event Rules. The logic that is attached to a table is run whenever any application initiates that database event. For example, to maintain referential integrity, you might attach rules to a master table that delete all children when a parent is deleted. Any application that deletes information from that table does not need to have the parent/child logic embedded in it because that logic exists in the table.

See [JD Edwards EnterpriseOne Tools Event Rules Guide](#) .



## Understanding System Functions

System Functions are procedures provided by the tool and are usually specific to the type of component being used. For example there are system functions to hide and show fields on an application, and there are system functions to execute different sections in a batch application. Parameters are passed into and out of the component for functionality. The available system functions are determined by the object type.

See *"System Functions in Form Design Aid" in the JD Edwards EnterpriseOne Tools Form Design Aid Guide* .

## Understanding Table I/O

Use the Table I/O button in Event Rules Design to create instructions that perform table input and output (I/O) so that you do not need to manually code a business function in C code. Table I/O allows you to access a table through event rules. You can use table I/O to do the following:

- Validate data
- Retrieve records
- Update or delete records across files
- Add records

For example, you can use table I/O to display information in a table that your application does not use. You can use Log Viewer to view your table I/O SQL statements in the `jdedebug.log`. To do so, your `jde.ini` file must have debugging set to File.

See *"Working with Table Input, Output" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .

## Understanding Business Function Design

This topic discusses both C business functions and named event rules, and includes information about master business functions, Business Function Builder, and business function documentation.

You can use business functions to enhance EnterpriseOne applications by grouping related business logic. Journal Entry Transactions, Calculating Depreciation, and Sales Order Transactions are examples of business functions.

You can create business functions using one of the following methods:

- Event rules scripting language  
The business functions that you create using the event rules scripting language are referred to as Business Function Event Rules (also called Named Event Rules). If possible, use Business Function Event Rules for your business functions. In some instances, C business functions might better suit your needs.  
**Note:** NERs get generated into either C or Java.
- C programming code

EnterpriseOne software creates a shell into which the user inserts logic using C. You use C business functions mainly for caching, but they can also be used for the following:

- Batch error level messaging
- Large functions

C business functions work better for large functions (as determined by the group). If you have a large function, you can break the code up into smaller individual functions and call them from the larger function.

- Functions for which performance is critical
- Complex Select statements

After you create business functions, you can attach them to EnterpriseOne applications to provide additional power, flexibility, and control.

See [JD Edwards EnterpriseOne Tools Development Standards for Business Function Programming Guide](#) .

## Understanding Processing Options

Processing options control how an interactive or batch application processes data. You can use processing options to change the way in which an application or a report appears or behaves. You can attach unique processing options to different versions of the same application, which allows you to change the behavior of an application without creating a new application. In addition, you can use processing options to do the following:

- Control the path that a user can use to navigate through a system.
- Set up default values.
- Configure an application for different companies or different users.
- Control the format of forms and reports.
- Control page breaks and totaling for reports.
- Specify the default version of a related application or batch process.

You can define processing options for an application that automatically appear at runtime. In addition, you might need to create a processing option version. The procedures for creating a processing option version are similar to those for creating an interactive version.

See ["Using Processing Options" in the JD Edwards EnterpriseOne Tools Data Structure Design Guide](#) .

# 6 Additional Topics

## Understanding Caching

Caching is a process that stores a local copy of frequently accessed content of remote objects. Caching can improve performance. EnterpriseOne software caches information in the following two ways:

- The system automatically caches some tables, such as those associated with constants, when it reads them from the database at startup. It caches these tables to a user's workstation or to a server for faster data access and retrieval.
- Individual applications can be enabled to use cache. JDECACHE APIs allow the server or workstation memory to be used as temporary storage.

JDECACHE can hold any type of indexed data that your application needs to store in memory, regardless of the platform on which the application is running; therefore, an entire table can be read from a database and stored in memory. No limitations exist regarding the type of data, size of data, or number of data caches that an application can have, other than the limitations of the computer on which it is running. Both fixed-length and variable-length records are supported. To use JDECACHE on any supported platform, you need to know only a simple set of API calls.

See *"Working with JDECACHE" in the JD Edwards EnterpriseOne Tools APIs and Business Functions Guide* .

## Understanding Messaging

Use EnterpriseOne messaging features to communicate pertinent information to the end user in the most effective and user-friendly way. When you design an application to use messaging, you must evaluate what information is necessary to enable a user to accomplish a task. You can deliver a message in real time, whereupon the message is displayed in an interactive application, or you can send messages to the Employee Work Center. The method that you use to provide information to the user depends on the situation. For example, you can do the following:

- Use an interactive error message if the system encounters an error during the entry of a record.
- Use an informational message that the system sends to the Workflow Center if information needs to be conveyed and responded to.
- Use an alert message if information is urgent and requires immediate attention.
- Use a batch error message if the system detects errors in a batch process, such as while a report is running.

The three components of creating system-generated messages are as follows:

- The message itself.

Do you require a simple message or a text substitution message? Are all of the text substitution pieces available?

- The logic that applies to the message.

Has certain criteria, such as event rule logic, been met so that a message should be sent?

- The message type.

Does the message require action by the users? Are all of the required parameters available at the time the message is to be sent?

## Batch Error Messages

The error message system gives users a consistent interface to review errors when working with batch programs. When a batch program has finished processing all messages regarding the success or failure of a job, the system sends a message to the user in the Employee Work Center. To enhance the usability of the messages, the system uses a tree structure (or parent/child structure) to group related messages. To provide additional flexibility and functionality, you can use text substitution, and you can make a message active, meaning that the user can open an associated form by clicking within the message.

The Employee Work Center displays the error messages that appear after a batch job has completed. When you create these batch error messages, you need to determine the possible messages EnterpriseOne users will need. For example, you might create a number of different messages that are generated when a journal entry report is run. You can create a message stating that the report completed normally if the report balances. Additionally, you can create multiple levels of messages describing various errors if the report is out of balance. The first level might state that the report completed with errors, and additional levels would explain the specific details about the errors.

See [JD Edwards EnterpriseOne Tools Batch Versions Guide](#) .

## Understanding Transaction Processing

A transaction is a logical unit of work (comprised of one or more SQL statements) performed on the database to complete a common task and maintain data consistency. Transaction statements are closely related and perform interdependent actions. Each statement performs part of the task, but all of them are required for the complete task.

Transaction Processing ensures that related data is added to or deleted from the database simultaneously, thus preserving data integrity in your application. In transaction processing, data is not written to the database until a commit command is issued. When this happens, data is permanently written to the database.

For example, if a transaction comprises database operations to update two database tables, either all updates will be made to both tables, or no updates will be made to either table. This condition guarantees that the data remains in a consistent state and the integrity of the data is maintained.

Using the EnterpriseOne development tools, you can enable an application for transaction processing and define which database operations comprise a transaction.

See ["Working with Transaction Processing" in the JD Edwards EnterpriseOne Tools Form Design Aid Guide](#) .

## Understanding Currency

Enterprises that do business internationally have additional accounting needs and added complexity. This complexity arises from doing business in different currencies and having to follow different reporting and accounting requirements. Some fundamental requirements for an international enterprise include:

- Conversion of foreign currencies to the local currency

- Conversion of the different local currencies into one currency for reporting and comparisons
- Adhering to regulations defined in the countries of operation
- Revaluation of currencies due to fluctuation in exchange rates

EnterpriseOne currency implementation is controlled by the developer and includes:

- Currency retrieval through database triggers and table event rules (TER).
- Business function event rules.
- System APIs for accessing cached tables.

See *"Using Currency" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .

## Understanding Media Objects

EnterpriseOne media objects and imaging features enable you to attach useful information to an application, including information that might currently exist as a paper-based document. The media objects feature enables you to attach the information to applications, forms and rows, and Object Librarian objects. The imaging feature within media objects gives you flexibility to create a more efficient method of information storage.

This table describes the types of information that you can attach to a grid row or a form:

| Type                                       | Description   |
|--|---|
| Text                                       | Media objects provide a word processor that lets you create a text-only attachment. For example, you can use a text attachment to provide specific instructions for a form or additional information about a record.  |
| Image                                      | Images include files such as Windows bitmaps, Graphics Interchange Format (GIF) files, and JPEG files. These files might represent electronically created files, as well as scanned images of paper-based documents.  |
| OLE  | Media objects can be files that conform to the OLE standard. OLE enables you to create links between different programs. By using these links, you can create and edit an object from one program in a different program. EnterpriseOne provides the links that you need to attach OLE objects. |
| Shortcuts                                  | A shortcut is a link that opens an EnterpriseOne application. Within media objects, you can only attach EnterpriseOne shortcuts; that is, you cannot attach Windows shortcuts to media objects.   |
| Uniform Resource Locations (URL) and files | Media objects can be links to web page URLs or other related files.   |

System administrators can also set up templates. A template might include attachments of its own, such as images and shortcuts. For example, you can create a letterhead and a standard form for a memo. You might create a shortcut in the template to provide access to an application that uses data specific to the information that you add to the template.

See *"Media Object Data Structures" in the JD Edwards EnterpriseOne Tools Data Structure Design Guide* .

See *"Understanding Media Object Processing" in the JD Edwards EnterpriseOne Tools System Administration Guide* .

## Understanding Debugging

Debugging is the method you use to determine the state of your program at any point of execution. You can use debugging to help you solve problems and to test and confirm program execution.

You can use a debugger to stop program execution so you can see the state of the program at a specific point. This allows you to view the values of input parameters, output parameters, and variables at the specified point. When program execution is stopped, you can review the code line-by-line to check such issues as flow of execution and data integrity.

You can use the following two tools for debugging EnterpriseOne:

- EnterpriseOne Event Rules Debugger
- Microsoft Visual C++ Debugger

You use the Event Rules Debugger to debug event rules and the following:

- Interactive applications
- Reports
- Table conversions

You use the Visual C++ Debugger to debug C business functions or NERs that are generated into C.

See *"Understanding the Cross Reference Facility" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .

## Understanding the Cross Reference Facility

You can use the Cross Reference Facility to determine where specific objects are used. You can also view relationships between objects and their components. For example, you can use 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.

Because the cross-reference files are not automatically rebuilt when objects are created and modified, the Cross Reference Facility should be rebuilt periodically. You can regularly schedule cross-reference builds to ensure that the cross-reference information is up-to-date.

See *"Using Cross Reference Facility" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .