
JD Edwards EnterpriseOne Tools 8.97 Development Tools: Report Design Aid Guide

October 2007

Understanding Batch Error Messaging.....	275
Setting Up Batch Error Messages.....	275
Understanding Level Break Messages.....	276
Understanding Level Break Message Components.....	278
Sample Source Code.....	280
Understanding Work Center APIs.....	283
Prerequisites.....	284
Creating Data Items for Level Break Messages.....	284
Creating Business Function Data Structures.....	285
Creating Level Break Message Business Functions.....	285
Calling the Work Center Initialization API.....	287
Calling the Processing Work Center APIs.....	287
 Glossary of JD Edwards EnterpriseOne Terms.....	 289
 Index	 305

Step	Reference
3. Set up permissions to create OMW projects.	<i>JD Edwards EnterpriseOne Tools 8.97 Object Management Workbench Guide</i> , “Configuring User Roles and Allowed Actions,” Setting Up Allowed User Actions
4. Set up activity rules to allow you to promote projects in OMW.	<i>JD Edwards EnterpriseOne Tools 8.97 Object Management Workbench Guide</i> , “Configuring Activity Rules”
5. Set up save locations to enable you to save JD Edwards EnterpriseOne objects that are not ready to be checked in.	<i>JD Edwards EnterpriseOne Tools 8.97 Object Management Workbench Guide</i> , “Configuring Object Save Locations”
6. Set up default locations and printers.	<i>JD Edwards EnterpriseOne Tools 8.97 Development Tools: Report Printing Administration Technologies Guide</i> , “Working with Report Printing Administration”

- Processing options

You can override several report specifications at the version level, including:

- Section layout
- Section data selection
- Section event rules
- Section database output
- Section sort sequence

Batch versions enable you to preserve template integrity while providing custom processing to meet a specific business need. Instead of creating separate report templates for multiple variations of a report, you can create one report template and add multiple batch versions.

See *JD Edwards EnterpriseOne Tools 8.97 Development Tools: Batch Versions Guide*

4. Perform one of these actions to refresh the preview:
 - Select Refresh Preview Window from the View menu.
 - If the system displays the View toolbar, click the Refresh Preview Window button.
 - Press F5 on the keyboard.
5. To modify the number of records that are processed and displayed in Preview mode, select User Options from the View menu.
6. On the User Options form, complete the Rows to Preview field to indicate how many table records to process in the preview.

PART 2

Enhancing Reports Using Basic Functionality

Chapter 5
Configuring the Report Design Aid Workspace

Chapter 6
Working with Report, Section, Field, Column, and Row Properties

Chapter 7
Working with Objects in Report Sections

Chapter 8
Working with Objects Unique to Tabular Sections

Chapter 9
Modifying the Appearance of Report Sections

Chapter 10
Modifying the Appearance of Report Objects

Chapter 11
Including Attachments and Comments in Reports

Chapter 12
Inserting Header and Footer Sections

Chapter 13

Working with Level Break Sections

Chapter 14

Working with Smart Fields

Note. Because only detail sections of a report have business views attached, the Business View Columns Browser is populated only when you select a detail section with an attached business view. Selecting other sections results in an empty browser window.

3. Drag fields from the Business View Columns Browser to the selected detail section.

Showing and Hiding the Data Dictionary Browser

Open a report in RDA.

1. Click a section in the report.

Note. Data dictionary items can be placed into any section type; therefore, you do not need to select a detail section.

2. From the View menu, select Data Dictionary Browser.
3. Use the QBE (query by example) line to search for an appropriate data item.
4. Drag the data item from the Data Dictionary Browser to the report section.

Icon	Field Type
XXX	Represents constant fields.
Dual-shaded box	Represents columns. If the arrow points to the top section, the icon refers to the column heading. If the arrow points to the bottom section, the icon refers to the column variable.
Gray, black, and white diamond	Represents runtime fields, such as report date, report time, page number, company title, and report title.
Multicolored diamond	Represents variables that are located in group sections.

Understanding Field Properties

Column headings and column variables in columnar sections as well as constants and variables in group sections are all considered fields where properties are concerned.

Field properties vary depending on the field type that is selected. Field properties are accessed from the Item Properties menu item and include these tabs:

Properties Tab	Description
General	Enables you to override the variable name of the field as it appears in Event Rules Design. Also enables you to override the column heading as it appears on the report. If the column heading is long, you can split it into two lines using the Col Heading 1 and Col Heading 2 fields.
Description	Enables you to override the name of the field as it appears in Event Rules Design. You can also view the name of the table where the field resides and view the field's data dictionary alias and name.
Font/Color	Enables you to modify fonts, font styles, font size, and font colors.
Style	Enables you to select line styles for the field.
Display	Enables you to modify the justification of the field, display length, display decimals, and edit code. The options that are available on this tab vary depending on the field type that is selected.
Totaling	Enables you to define a quick in-section total of the field. This tab is available for numeric fields only.
Decimal Scaling	Enables you to apply decimal scaling to numeric fields. This tab is available for numeric fields in tabular sections only.
Advanced	Enables you to select advanced options.

Advanced Field Properties

The advanced field properties are typically the same for every field type.

1. Click the variable of a numeric field.
2. From the Edit menu, select Item Properties.
3. On the properties form, select the Totaling tab.
4. Select one or both of these options:
 - Total
 - Grand Total
5. Select from these aggregate functions, and click OK:
 - Sum
 - Average
 - Minimum
 - Maximum
 - Count
6. From the File menu, select Report Properties.
7. On the Report Properties tab, select one or both of these options:
 - Print Totals Only
 - Print Grand Totals

These two options correspond to the options that you selected in step 4.

Note. Depending on the options that you selected, the system adds one or two lines to the bottom of each column before each section break. If you selected both the total and grand total options, the total line appears before the grand total line. The totals are not labeled. In fact, except for the column that displays the total, information from the last record is repeated in the total line. You can suppress totals for fields that you do not want to appear on this line.

CHAPTER 8

Working with Objects Unique to Tabular Sections

This chapter provides an overview of tabular sections and discusses how to:

- Work with Row Description columns.
- Define decimal scaling.
- Create calculation columns.
- Work with rows.
- Override cells and cell properties.

Understanding Tabular Sections

Tabular sections are specialized types of columnar sections. While the report data is presented in a columnar format, tabular sections provide spreadsheet capability by enabling you to define the data in columns, rows, and cells.

When you include fields that display numeric values, tabular sections automatically total the values. For example, if you include an object that displays open amounts, tabular sections calculate a grand total of all the open amounts in the section. Because totaling is automatic in tabular sections, level break footers are not available. You must suppress totals for any columns that you do not want calculated.

In addition, level break headers are not available in tabular sections. Use the Row Description column to describe the data.

When you create tabular sections, observe these guidelines for defining combinations of columns, rows, and cells:

Condition	Guideline
Define columns only.	Define columns when the information in the report is based solely on the data that is contained in tables. When you define the columns, rows are generated at runtime based on the selection, sequence, and level break criteria that you defined.
Define columns and rows.	Define rows in addition to columns when you include details in the report, such as underlines, spaces, and blank lines, as well as special calculations, such as interim totals. Row information is set up horizontally on the report.
Define columns, rows, and cells.	Define cells to override information that is defined by columns and rows. A cell is the intersection of a column and row.

4. From the Cell menu, select Create Override, Constant.
5. On the Cell Properties form, modify the name and description on the General tab, if appropriate.
6. Set properties on the available tabs as appropriate, and click OK.

Adding, Modifying, and Deleting Comments

Open a report template in RDA.

1. Right-click the object to which you want to attach the comment, and select Insert Comment.
2. Enter the text, and then click anywhere in the design workspace to close the text window.
3. Right-click the object and modify, delete, show, or hide the comment.

3. Select fields from the Insert menu, and drag them to the report footer section.

You can add any type of data field except business view columns and the Page n of Total runtime field.

4. Double-click the Report Footer section to open the Report Footer properties form.
5. Define section properties as appropriate, and click OK.

2. Perform one of the following actions:
 - If the detail section is a columnar or group section, select Smart Field from the Insert menu.
 - If the detail section is a tabular or application report section, select Create, Smart Field from the Column menu.
3. On the Create New Smart Field form, select a smart field.

The smart field director guides you through the process of setting up the smart field.
4. To move the column, drag it to the new location.
5. To delete the column, click it and select Delete from the Edit menu.

PART 3

Enhancing Reports Using Advanced Functionality

Chapter 15
Understanding Advanced Report Enhancements

Chapter 16
Joining Detail Sections

Chapter 17
Including Text Attachments in Reports

Chapter 18
Working with the Drill Down Feature

Chapter 19
Working with Database Output

15. On the Event Rules Design form, click the check mark to save the event rules and return to RDA.

16. Preview the report.

For any record that has generic text attached, the text prints in the alpha variable that you placed on the report.

2. Select a file, and click Delete.

PART 4

Understanding Logic and Processing

Chapter 20
Working with Event Rules

Chapter 21
Understanding Events

Chapter 22
Understanding Report Processing

Chapter 23
Understanding Runtime Processing

Chapter 24
Working with Report Interconnects

3. On the Text Variables form, enter the text that you want to appear on the report under the Text String heading.
4. Press ENTER or click Add, and then click OK.

You must press ENTER after each entry for RDA to recognize the entry. When you press ENTER or click Add, another blank line appears.

Using Text Variables in Assignments

Open a report template that includes text variables in RDA.

1. Click the section to which you added the text variables, and select Event Rules from the Edit menu.
2. On the Event Rules Design form, select an event from the events drop-down list box.
3. Click the existing statement to add a statement to follow it.
4. Create an assignment by selecting Assignment/Expression from the Insert menu.
You can also click the x= button on the toolbar.
5. On the Assignment form, select the field to which you want to assign a value from the To Object list.
6. Select the text variable in the From Object/Literal field, and click OK.

Refer to the prefix code table in the Understanding Event Rules section.

The Event Rules Design form displays the assignment using the text variable in addition to any existing event rules.

7. Click the check mark on the toolbar to save and quit Event Rules Design.

Calling System Functions in Event Rules

This section provides an overview of system functions, lists the prerequisites, and discusses how to use system functions in event rules.

Understanding System Functions

System functions are predefined sets of logic that are shipped with the JD Edwards EnterpriseOne product. These functions enable you to perform specialized processing without adding custom code. This table shows some examples of frequently used system functions:

System Functions	Description
Hide Object and Show Object.	Enables you to hide and show objects within a section.
Hide Section and Show Section.	Enables you to hide and show sections within a report.
Do Custom Section.	Enables you to call a custom section.
Set Selection Append Flag and Set Sequence Append Flag.	Enables you to set append flags for defining data selection and data sequencing entered into processing option templates.

Custom Section Usage	Description
Perform logic.	Create a custom section and add event rules to perform percent of total calculations to be used by the level-one section.
Present additional information when stated criteria is met.	Create an accounts receivable report that displays the payment history of customers. Add a custom section that presents account receivables by aging categories. Call the custom section <i>only</i> if a customer is delinquent. When the batch engine encounters a record that meets the past due criteria, the custom section prints. The custom section does not print for any records that do not have a past due amount.

Logic for Custom Sections

When you use a custom section, you must specify the section as conditional in the section properties. Use the Do Custom Section system function to call the custom section. You can use the Do Custom Section system function in columnar, group, or tabular sections. You must attach the system function to the section that precedes the custom section. For example, to process a custom section after a columnar section, call the custom section from the Do Section event of the columnar section. Likewise, to process a custom section after a level break footer, call the custom section from the Init Lvl Brk Footer Section event.

You can call a custom section from any report section. Furthermore, you can use any event rule logic along with a custom section, such as If/While statements, business functions, and table I/O.

You can call a custom section from any event except the Initialize Section event. If you call a custom section from the Initialize Section event, the report does not process.

Prerequisites

Before you begin creating custom sections, ensure that you:

- Create a batch application object.
- Complete the design of the report template.

Setting Up Custom Sections

Open a report template in RDA.

1. Click the section from which you want to call the conditional section.
2. From the Edit menu, select Event Rules.
3. Select the Do Section event from the events drop-down list box.
4. From the Insert menu, select If/While and define the criteria for displaying the conditional section.
5. Click the If/While statement, and select System Function from the Insert menu.
6. On the Function Selection tab, expand the Section folder and double-click Do Custom Section.
7. On the Parameter Mapping tab, select the report section that you want to display when the defined criteria is met.

The section appears in the Values column of the Parameters table.

8. Click OK, and save the event rules.

9. In the RDA workspace, double-click the report section that you selected to appear only when the defined criteria is met.
10. On the appropriate section properties form, select the Advanced tab.
11. Select the Conditional option, and click OK.

Accessing BrowsER for Report Templates

This section provides an overview of BrowsER, lists the prerequisites, and discusses how to access BrowsER.

Understanding BrowsER

Use Oracle's JD Edwards EnterpriseOne BrowsER to view event rules for an entire report or batch version. JD Edwards EnterpriseOne BrowsER displays the report sections in a hierarchical structure. When you expand a section, you see a section node and nodes for each object within the section. Event rules that are attached to the section appear under the section node. All event rules that are attached to an object appear under the appropriate object node.

A plus sign next to a node indicates that event rules are attached. When you expand a node with a plus sign, an event node on which the event rules are attached appears with a plus sign. When you expand the event node, you can review the event rules that are attached to the event. You can disable and enable one or more event rules from JD Edwards EnterpriseOne BrowsER. This is useful for troubleshooting event rules for an entire report or batch version.

You can access JD Edwards EnterpriseOne BrowsER from multiple locations in JD Edwards EnterpriseOne:

- Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).
Open a report template in RDA, and select BrowsER from the View menu.
- Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW).
Select a report template, and access the Design Tools tab.
- Oracle's JD Edwards EnterpriseOne Batch Versions.
Select a batch version, and from the Advanced Operations form, select Report BrowsER from the Form menu.

See Also

JD Edwards EnterpriseOne Tools 8.97 Development Tools: Batch Versions Guide, "Reviewing Batch Version Processing," Accessing BrowsER for Batch Versions

JD Edwards EnterpriseOne Tools 8.97 Development Tools: Event Rules Guide, "Using BrowsER"

Prerequisites

Before you access JD Edwards EnterpriseOne BrowsER for report templates, ensure that you:

- Create a batch application object.
- Complete the design of the report template.
- Include event rules in the report template.

Accessing BrowsER

Access OMW.

1. From a project, select the report template that you want to review.
For example, select the R014021 One Line Per Address report template.
2. Click Design.
3. Select the Design Tools tab.
4. Click Browse Event Rules.
5. On the Browsing form, expand the nodes for each report section.
If no other nodes under the report section can be expanded, no event rules exist for that report section.
6. Expand additional nodes until you locate event rules.
You can disable and enable event rules for the entire report template from this form.

Event	Report Header	Page Header	Columnar	Group	Tabular	Level Break Header	Level Break Footer	Page Footer	Report Footer
Do Column Heading	N	N	Y	N	Y	N	Y	N	N
Do Constant	Y	Y	Y	Y	Y	Y	Y	Y	Y
End Constant	Y	Y	Y	Y	Y	Y	Y	Y	Y
Initialize Const	Y	Y	Y	Y	Y	Y	Y	Y	Y

See Also

JD Edwards EnterpriseOne Tools 8.97 Development Tools: Event Rules Guide, “Using Event Rules Design”

Each section type includes different events; therefore, the event flow for each section type is slightly different. The events for each section type are described in the Event Levels section of this chapter.

Group and Columnar Section Event Flow

The system processes group and columnar sections alike because they both write to the output after each record is read. This processing is unlike tabular section processing, which writes to the output only when a level break is encountered. The output is determined by the data selection and data sequencing defined in the section.

This diagram illustrates the typical event flow for group and columnar sections:

If an alternative description exists for business view favorites folders, subfolders, and business views in the specified language, they appear in that language. All other business view favorites folders, subfolders, and business views appear in the domestic language. The notes are blank to enable you to also add notes in the alternative language.

Description	The description of the data items to be used as data sequencing and level breaks in the report sections created using the Report Director template. The system populates the description field based on the alias that you enter.
Display Seq	Indicates the sequence in which the data sequencing fields appear on the Data Sequencing Help form of the Director. The system populates this field when you tab out of it.

See Also

Chapter 29, “Creating Report Director Templates,” page 251

Designing Reports Using Custom Smart Fields

Access RDA.

1. On the Director, under Application Reports, select the appropriate Report Director template from the pull-down menu and click Next.

This information is derived from the description you entered when you created the Report Director template.

2. On the Page Header Details form, accept the page header defaults and click Next.
3. On the Business View Selection Options form, select the option I'll use the pre-defined business view and click Next.

The predefined business view is the business view that you defined on the Building Blocks tab for the Report Director template. You can select one of the other two business view selection options to override this default.

4. On the Select Columns form, move the appropriate smart field from the Available Smart Fields list to Columns in Report Section.

The available smart fields are derived from the Smart Field template that you associated with the Report Director template. All smart fields in the attached Smart Field template are available to the report developer.

5. On the Smart Field Name form, change the Variable Name to describe the smart field (for example, First Quarter).

- Change the Report Column Headings fields to describe the field on the report.

For example, to describe the first quarter data, enter *First Quarter* on line 1 and *Sales* on line 2.

If you selected a Report Director template that includes a smart field that displays column headings, the Smart Column Heading option is available. Instead of changing the column heading names, select this option to enable the smart field to populate the column headings based on information that you enter in the subsequent forms.

Subsequent forms are dependent on the selected Report Director template. For example, using the quarterly sales scenario you might see these forms:

- Smart Field Parameters

This form displays the user prompt data item that you created. The information that you entered in the item glossary appears in the center of the form. Enter an appropriate value in the Enter the Literal field based on the glossary information. If you created multiple user prompts, you are presented with additional forms for each prompt. For quarterly sales, enter the quarter that you want calculated and included on the report. For first quarter sales, enter *1* in this field.

- Smart Field Data Selection

This form displays the data items that you defined on the Smart Field Template Criteria Revisions form when adding the smart field to the smart field template. If you defined any of the data items as requiring a range of values, the data item displays a From and a To field.

If you intend to define the same data selection for each smart field column, you can leave this form blank and just fill in the data selection for the report.

6. When you have completed all smart field parameters, click Finish.

Add additional smart field columns to the report. For example, for quarterly sales, you can select the smart field again and enter 2 on the Smart Field Parameter form to calculate and display sales for the second quarter.

7. When you are finished adding smart field columns to the report, click Next.

8. On the Data Sequencing Help form, click Next.

The Data Sequencing Help form displays the Data Sequence and Level Break Fields that you defined in the Report Director template. The fields that appear under Report Grouping are defined as level break fields and are the first two fields you entered into the detail area in the Report Director template. You can remove these fields as level breaks by clearing the option.

The fields that appear under Report Detail are the additional fields that you included in the Report Director template and are also used for data sequencing. Click each field that you want to include as a level break field so that they appear in the empty field directly above them.

You can override the predefined data sequencing and level breaks by selecting the I'd like to set up the sequencing and level breaking myself option under Advanced. You can define data sequencing on the Section Data Sequencing form. The fields listed in the Available Columns section are fields from the attached business view. The subsequent Define Sort Properties form enables you to define level breaks.

9. On the Help with Section Data Selection form, create either a balance sheet or an income statement and indicate whether you want to add to the data selection, and click Next.

Note. This form only appears if you defined the associated Report Director template to display financial criteria.

10. On the Data Selection form, define the appropriate data selection.

This form appears if you defined the selected Report Director template to display generic criteria or if you selected the option Set up data selection manually on the Help with Section Data Selection form.

You should define data selection on the Data Selection form to enhance system performance *even if* you defined data selection for each smart field.

11. On the Additional Properties form, select the appropriate options and click Next.

The options that appears on this form are defined on the Properties tab of the Report Director Template.

12. On the Finish form, click Finish.

13. Save and preview the report.

CHAPTER 31

Defining Batch Error Messages

This chapter provides an overview of batch error messaging and discusses how to set up batch error messages.

Understanding Batch Error Messaging

The Oracle's JD Edwards EnterpriseOne error message system provides a consistent interface to review batch application errors. You can set up batch applications to send messages to users when processing is complete. These messages include the success or failure of the batch job and notification when information is incorrectly entered into the system. To enhance the usability of the messages, the system uses a tree structure (or parent/child structure) to group related messages.

You can create two types of error messages for batch applications:

- Non-text substituted
- Text substituted

A non-text substituted error message provides a high-level indication that an error exists in the batch application. For example, *Document number is invalid*.

You can use text substitution messages to provide additional information regarding errors. For example, rather than the *Document number is invalid* error message, the message can instead state *Document number 55.5555 is invalid*.

You can also include a link to the associated interactive application in the message. This type of message is referred to as an *active* error message. The user can open the associated interactive application from the link that is attached to the message.

Error messages appear in the Oracle's JD Edwards EnterpriseOne Work Center (P012503) application after the batch job has completed. When you create custom batch error messages, determine what information Oracle's JD Edwards EnterpriseOne users require. 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 successfully when the report data is in balance. Additionally, you can create multiple levels of messages that describe various errors if the report is out of balance. The first level might state that the report completed with errors while additional levels explain the specific details regarding the errors.

Setting Up Batch Error Messages

This section provides overviews of level break messages, level break message components, and work center APIs, provides sample source code, lists the prerequisites, and discusses how to:

- Create data items for level break messages.

- Create business function data structures.
- Create level break message business functions.
- Call the work center initialization API.
- Call the processing work center APIs.

See Also

JD Edwards EnterpriseOne Tools 8.97 Development Standards for Business Function Programming Guide, “Implementing Error Messages”

Understanding Level Break Messages

Level break messages act as containers to present error messages for each level break field that is defined in the batch application. Level break messages are informational; they are not error messages. Level break messages can be action messages that contain a shortcut to an application and require action on the part of the user. They can also be messages that require no action, but might include instructions for the user to review information.

Level break messages are text-substituted messages. You can define them at many different levels such as level one, level two, level three, and so on.

The level one level break message appears when you first launch JD Edwards EnterpriseOne Work Center. Additional levels appear beneath this level. This level of message provides high-level information. For example, *Job completed with errors*.

The level two level break message appears when you expand the level one message. Additional levels can appear beneath this level. This level of message provides more detail. For example, *Batch 3230 has errors*.

The level three level break message appears when you expand the level two message. Additional levels can appear beneath this level. This level of message provides additional detail. For example, *Voucher 14787 contains errors*.

How Level Break Messages Work

Level break messages group errors. They include these two distinct components:

- The actual text of the message.
- An indication whether the message is an action message.

All level break messages include the text component, but not all messages are defined as action messages.

The Work Center API creates and manages the Job Completed level break message. You create custom messages to meet the needs of your company. Level break messages organize messages that you set up to communicate information about the batch, document, and line in which the error occurred. You set these messages using the `jdeSetGBRError` or the `jdeSetGBRErrorSubText` functions.

Level break message *Job R89004 ZJDE0001.c* is a level-one message that the system generates automatically. Level break message *Batch 3230 is in Error* is a level-two message. *AA 20050606 3031* and *Intercompany Out of Balance* are actual error messages.

The Work Center API must be called for every level break message.

These topics illustrate how messages might appear when an out-of-balance journal entry upload has completed.

First-Level Messages

First-level messages appear when users open their personal in-basket in Work Center. A plus symbol next to the message indicates that additional levels of messages exist. First-level messages might show the name of the batch job, explain that it completed with errors, and instruct the user to review the details about the errors.

Second-Level Messages

Users can view second-level messages when they expand the first-level message in JD Edwards EnterpriseOne Work Center. The second-level message might inform users that they need to review a specific batch number.

Third-Level Messages

Third-level messages appear when the user expands the second-level message in JD Edwards EnterpriseOne Work Center. The third-level message might inform the user that the batch job completed with errors because it was out of balance and then provide solutions to resolve the issue.

Text Substituted Messages

Error messages must be informative to be beneficial. You can increase their effectiveness through text substitution. Text substitution enables you to define variable text in messages (such as dates, amounts, and so on) that the system replaces with data at runtime. You set up text substitution messages in the data dictionary, which helps to ensure consistency of jargon and terminology. For example, in the message *Voucher Batch &1 Contains Errors*, the system uses *&1* as a parameter to the message. The system substitutes *&1* at runtime with a batch number and stores the rest of the information from the message in the data dictionary glossary. The glossary describes the error message as it is defined in the data dictionary. When you open the message in data dictionary, you can review the item glossary.

Action Messages

After users review a an error message and determine the resolution, they typically need to access an application to resolve the error. You can set up specific level break messages, which are known as action messages, to provide direct access to the associated application directly from JD Edwards EnterpriseOne Work Center. Action messages call a JD Edwards EnterpriseOne application and pass the required variables to that application. You determine the appropriate application and the correct values that need to pass to that application when you create the message. The system highlights action messages in the detail area to differentiate them from non-action messages. For example, users can click the shortcut to automatically access the Voucher Revision form from within the error message. The application presents the appropriate form and displays the record in error.

Work Center APIs

When you call a Work Center API, the system assumes a child/parent order. In other words, when the API is called, the system assumes that any error that is in the runtime error message stack belongs to the level that is associated with that instance of the API call. This means that all of the errors in the error space at that time, whether they are set using business functions or event rules, are packaged or grouped together as children of the level that was passed to the Work Center API. These error messages are then cleared from the error space so that the system can create the next group of messages based on a new set of records.

The timing of the calls to the Work Center API is critical. The reporting program typically starts by editing the header-level record, which leads to a set of detail records. The detail records are the first to be read and processed. Thus, the calls to the work center API typically send level break numbers in descending level break order.

For example, the actual series of level break calls to the API might appear as 4, 4, 4, 3, 4, 4, 3, 2, 4, 4, 3, 2, 1. This series indicates that the call structure started four levels down. The first call at level 4 allows the Work Center API to find any messages that occurred at that time and creates child messages using the level 4 message as the parent. If no errors occurred, then no messages are created. This call sequence example illustrates that the API was called at level 3 after three calls to level 4. When the call to level 3 is made, the Work Center API remembers if any level 4 messages were written. In other words, if no errors occurred when any of the level 4 calls were made, then the Work Center API does not create the level 3 messages. If even one error existed at any of the level 4 calls, the Work Center API creates the level 3 and the level 2 messages.

You must call the Work Center API at every level. Because the work center error messages are created based on a parent/child structure, if a level call is skipped, then the API has no way to group the child messages and child levels that are already created.

For example, this level call structure is valid: 6, 6, 5, 4, 3, 4, 4, 3, 2, 1. Conversely, the call sequence 6, 6, 4, 3, 4, 3, 2, 1 is invalid because after level 6 is called, there is no corresponding call to level 5.

The Work Center API must be called using level 1 when the reporting job is about to complete. Hence, level 1 is the parent to all errors and level break messages. The level 1 call issues the job completed message. The level 1 call to the Work Center API is essential because it ensures that no orphan JD Edwards EnterpriseOne Work Center records are created and it also cleans up all allocated storage used by the Oracle's JD Edwards EnterpriseOne Work Center system. The level 1 call to the API should occur only once, typically in the End Section event of the primary section of the report.

Understanding Level Break Message Components

Error messages and level break messages are considered glossary data items. Level break messages act as a container for error messages.

You create level break messages in a project in Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW) by creating:

- Data dictionary items.
- Text substituted data structures.
- Business function data structures.
- Business functions.

Data Dictionary Items

The data dictionary item defines the text portion of the level break message as it appears in the JD Edwards EnterpriseOne Work Center. Before you create a data dictionary item, you should review the existing level break messages to determine if one meets your business needs.

When you create a custom level break message data item in the data dictionary, define it as a glossary data item. The alias that you enter for the data dictionary item is a unique identifier and cannot be changed once the data item is saved. It is recommended that the alias begins with LM. Level break messages are not error messages; they are defined as glossary group Y, PPAT Level Messages.

This description of the level break message data item appears in the JD Edwards EnterpriseOne Work Center. If this is a text-substituted message, enter the description using ampersands and sequential numbers to hold positions for substituted variables. For example, Batch &1 is out of balance by &2. The system replaces &1 with the actual batch number while the system replaces &2 with the amount that the batch is out of balance.

All message data items must include a cause and resolution. Begin by entering the text *Cause:* and then enter the cause of the message. Then, under the cause, enter the text *Resolution:* and then enter how the user should resolve the issue. For text-substituted messages, enter the data item description in the text area just above the cause.

Text Substituted Data Structures

The data structure for a text substituted error message must include the data items that are required for the text substitution. For example, a level break message describing a batch number that is out of balance by a specific amount uses a data structure that includes data items ICU (Batch Number) and AA (Amount).

The name of the data structure should include the same unique number that is used for the associated data dictionary glossary item. This number is to be appended to the prefix *DELM*. For example, a data structure created for level break message LM5509 is named DELM5509.

You must create a type definition (typedef) of the data structure to include in the associated business function. The typedef adds code to the C business function so that the business function can use the data structure.

Business Function Data Structures

You must have a business function for each level break message. Each business function requires an attached data structure. Do not confuse the business function data structure with the text substituted data structure that you created for the data dictionary item. The difference between the two is that the business function data structure moves data variables to the level break function. The data structure for the data dictionary items stores data that is mapped to the text substituted variables.

Include these items in the business function data structure:

- All data items that are required for the level break text substitution message.
- All data items that are required for the message to be active; that is, any variables that are required to load the form for the appropriate application.
- Data item EV01.

Change the variable name from *cEverestEventPoint01* to *cIncludeInterconnect*. This parameter is a flag that determines if the message is active. This parameter should be included as a parameter in all level break messages, even if the original intent is not to call an application. You must enter a 1 in the data structure value to launch an application.

- Data item GENLNG.

Change the variable name from *idGenericLong* to *idGenlong*. Use this parameter to control all JD Edwards EnterpriseOne Work Center messaging. This data item is intended for use as a work field for the system.

For the system to return the correct batch number and the amount that is out of balance, the business function data structure must include these fields:

- Document number
- Document type
- Document company
- Batch number
- Batch type
- Document pay item

Two additional data items are required in this data structure:

- J.D. EnterpriseOne Event Point (EV01).
Required to determine if the message is an action message.
- Generic Long (GENLNG).
Required to control JD Edwards EnterpriseOne Work Center messaging.

The name of this data structure should include the same unique number that is used for the associated data dictionary glossary item. This number is to be appended to the prefix *DLM*. For example, you should name a business function data structure that was created for level break message LM5509 as DLM5509.

Business Functions

After you have created the business function data structure, you can create the business function. The business function processes the level break errors and performs all of the mappings for the active message.

The name of the business function should include the same unique number that is used for the associated data dictionary glossary item and the data structures. This number is to be appended to the prefix *BLM*. For example, you should name a business function that you create for level break message LM5509 as BLM5509.

You must attach the business function data structure to the business function prior to entering code. Then, when you create the business function, you can have the system create a skeleton for you. The last message that you receive reminds you to create a *typedef* of the business function data structure and paste it into the header file of the business function.

After you create the business function, you must build it and check it in to central objects.

See Also

JD Edwards EnterpriseOne Tools 8.97 Development Standards for Business Function Programming Guide, “Implementing Error Messages,” Using Text Substitution to Display Specific Error Messages

Sample Source Code

You need to manually map fields from the business functions data structure to the `dsTextData` data structure; this is the data structure that is used for the text substitution in the level break message. You also need to manually map fields from the business function data structure to the `dsFormData` data structure; this is the data structure that is used for the active message.

This sample of the shell source code illustrates the information required in a level break message business function and its location in the `.c` file:

Variable Declarations

These lines declare the level break message variables:

```

/*****
* Variable declarations
*****/
JCHAR szForm[11]; /* Name of form in application */
JCAHR szDDitem[11]; /* Data dictionary name of the level message */
JCHAR szDLLName[11]; /* Name of the application DLL */
JCHAR szDs Tmpl[11]; /* Name of the text substitution data structure */

```

Declare Structures

Enter your own code for the appropriate text substitution data structure and application form. These examples are from an existing business function:

```

/*****
* Declare structures
*****/
DSDELM0002 dsTextData; /* Instance of text substitution structure */
FORMDSW0411Z1D dsFormData; /* Instance of form interconnect structure */

```

Set Pointers

These lines ensure that the level break message functions:

```

/*****
* Set pointers
*****/
if (lpDS->id⇒GenLong == (ID) 0)
{
    jdeSetGBRError (lpBhvrCom, lpVoid, (ID) 0, _J("4363"));
    if (hUser)
    {
        JDB_FreeBhvr(hUser);
    }
    return ER_ERROR;
}
else
    lpDswork = (LPDS_B0100011A) jdeRetrieveDataPtr(hUser, lpDS->idGenLong);

```

Main Processing

Enter your own code for the appropriate text substitution data structure and glossary data item. These examples are from an existing business function:

```

jdeStrncpy((JCHAR*) szDsTpl, (const JCHAR*) (_J("DELM0002")), DIM(szDsTpl)-1);
jdeStrncpy(szDDitem, (const JCHAR *) (_J("LM0002")), DIM(szDDitem));
memset((void *) (&dsTextData), (int) (_J("\0')), sizeof(dsTextData));

```

Assign Values from lpDS Data Structure to dsTextData Here

Enter your own code for the data items that are included in the business function data structure. When you assign values, map the business function data structure items to the text substitution data structure items. These examples are from an existing business function:

```

jdeStrncpy(dsTextData.szEdiuserid, (const JCHAR *) (lpDS->szEdiuserid),
    DIM (dsTextData.szEdiuserid));
jdeStrncpy(dsTextData.szEdibatchnumber, (const JCHAR *)
    (lpDS->szEdibatchnumber),
    DIM(dsTextData.szEdibatchnumber));
jdeStrncpy(dsTextData.szEditransactnumber, (const JCHAR *)
    (lpDS->szEditransactnumber),
    DIM(dsTextData.szEditransactnumber));

```

The first parameter is the location *to* where the value is being copied. The second value is the location *from* where the value is being copied. In this example, `dsTextData.szEdittransactnumber` is located in the text substitution data structure and `pDS->szEdittransactnumber` is located in the business function data structure.

Note. Be conscientious of the commands that you use for these statements. The commands are based on the data type of the associated data items. The `MathCopy` command is for math numeric fields, assignments are for character fields, `Strncpy` is for strings, and `Memcpy` is for dates. If you use the `Memcpy` command for dates, the system assigns the characters directly.

The remaining lines in this section ensure that the level break message is functional:

```
if (lpDSwork->lpBlob->lpTSDSMPL != (LPDSTMPL) NULL)
{
    lpDSwork->
        lpBlob->lpTSTEXT= (PJSTR) AllocBuildStrFromDstmplName ((LPDSTMPL)
            lpDSwork->lpBlob->lpTSDSMPL, (JCHAR*) szDsTmpl,
            (LPVOID) &dsTextData);
    jdeStrncpy (lpDSwork->lpBlob->szDDItem, (const JCHAR *) (szDDItem),
        DIM(lpDSwork->lpBlob->szDDItem));
}
if (lpDS->cIncludeInterconnect == _J('1'))
```

Form Interconnect Processing

Enter your own code for the appropriate application and form to be linked to the message. This example is from an existing business function:

```
jdeStrncpy (szDLLName, (const JCHAR *) (_J("P0411Z1")), DIM(szDLLName));
memset ((void *) (&dsFormData), (int) (_J'\0'), sizeof (dsFormData));
memset ((void *) (szForm), (int) (_J'\0'), sizeof (szForm));
jdeStrncpy ((JCHAR *) szForm, (const JCHAR *) (_J("W0411Z1D")), DIM(szForm) - 1);
```

Assign Values from LpDS Data Structure to dsFormData

Enter your own code for the appropriate application form. This example is from an existing business function and illustrates how to pass information from the business function data structure to the form data structure:

```
jdeStrncpy (dsFormData.EDUS, (const JCHAR *) (lpDS->szEduuserid),
    DIM(dsFormData.EDUS));
jdeStrncpy (dsFormData.EDBT, (const JCHAR *) (lpDS->szEdibatchnumber),
    DIM(dsFormData.EDBT));
jdeStrncpy (dsFormData.EDTN, (const JCHAR *) (lpDS->szEdittransactnumber),
    DIM(dsFormData.EDTN));
ParseNumericString (&dsFormData.EDLN, _J("1.0"));
dsFormData.EV01=_J('1');
```

Note. Be conscientious of the commands that you use for these statements. The commands are based on the data type of the associated data items. The `MathCopy` command is for math numeric fields, assignments are for character fields, `Strncpy` is for strings, and `Memcpy` is for dates. If you use the `Memcpy` command for dates, the system assigns the characters directly.

Get the Form Data Structure ID from the SVRDTL Table

These lines ensure that the level break message is functional:

```

If (JDESPECRESULT_PASSED==jdeSpecOpenLocalIndexed(&hTam,hUser,
    JDESPECTYPE_SVRDTL,SPECKEY2_SVRDTL))
{
    jdeStrncpy((JCHAR *)lpDWork->lpBlob->szForm,(const JCHAR *) (szForm),
        DIM (lpDWork->lpBlob->szForm)-1);
    JdeStrncpy(Key.szForm,szForm,DIM(Key.szForm));
    ASVRdtlData.DataType=JDESPECDATA_RAWBLOB;
    JdeSpecFetchSingle(hTam,&ASVRdtlData,&Key,1);
    If (ASVRdtlData.pSpecData !=(void *)NULL)
    {
        lpASVRdtl=ASVRdtlData.pSpecData;
        JDBRS_GetSTMPLSpecs(hUser,(JCHAR*)lpASVRdtl->szFITemplateName,
            &lpDWork->lpBlob->lpFIDSMPL);
        if (lpDWork->lpBlob->lpFIDSMPL !=(LPDSTMPL) NULL)
        {
            lpDWork->lpBlob->lpFITEXT=(PJSTR)AllocBuildStrFromDstmplName
                (LPDSTMPL)
            lpDWork->lpBlob->lpFIDSMPL,(JCHAR*)lpASVRdtl->szFITemplateName,
                (LPVOID)&dsFormData);
            jdeStrncpy(lpDWork->lpBlob->szDLLName,(const JCHAR *) (szDLLName),
                DIM(lpDWork->lpBlob->szDLLName));
        }
        jdeSpecFreeData(&ASVRdtlData);
    }
    jdeSpecClose(hTam);
}

```

Function Clean Up

These lines ensure that the level break message is functional:

```

if(hUser)
{
    JDB_FreeBhvr(hUser);
}
return (ER_SUCCESS);}

```

Understanding Work Center APIs

In the batch application, use API calls in event rules to:

- Initialize the Work Center.
- Identify the level break points.
- Terminate the batch message process.

Initializing the Work Center

In the batch application, initialize the JD Edwards EnterpriseOne Work Center API in event rules. You typically use the initialize section event of the primary section.

The `cAllowUserIdToChange` parameter on the initialize API works in combination with the `szUserid` parameter on the `ProcessErrorsToPPAT` API. The `cAllowUserIdToChange` parameter enables you to set up the batch application to send errors to the user who created the original records and not to the person who submits the job (such as the night operator). For example, if a single batch job contains 1,000 transactions that were created by 50 users, then only those users who created transactions with errors receive error messages. The night operator still receives a message, but it is a message such as *Job completed normally* or *Job completed normally with errors*. Other users whose transactions contained no errors do not receive error messages.

To set up this functionality, you need to enter a *1* in the `cAllowUserIDToChange` parameter when you initialize the batch error processing system. When you process the level 2 level break message and then call the `ProcessErrorstoPPAT` API, you can still specify who receives the messages by using the `szUserid` parameter. You can determine who should receive the message by reviewing the transaction record.

Identifying Level Break Points

After the JD Edwards EnterpriseOne Work Center system has been initialized, identify the various level break points within the report. Call the Processing Work Center API at each of these points to group the errors.

Terminating the Work Center Process

You must terminate the work center process before the batch job is finished but after all messages have been sent to the JD Edwards EnterpriseOne Work Center. When the batch program is about to terminate, call the JD Edwards EnterpriseOne Work Center error message business function, `ProcessErrorsToPPAT`, one last time. This call sends the process to level 1. Level 1 indicates the level of totaling is equal to 1 and that the process is complete. The system creates the job-completed message and frees any workspace that the Work Center API is using. This API is typically called on the End Section event of the primary section of the batch application.

Every report design that uses the Work Center API to process errors must call the API at the end of processing using a *1* in the level of totaling field. This call should also be done by jobs that are monitoring for critical errors and that need to terminate early.

When the system finishes processing the report, it creates messages, which can be reviewed in the JD Edwards EnterpriseOne Work Center. Batch errors are processed to the JDEM system. The system sends messages to the user running the report unless you specify that the messages should be sent to other users.

If the system encounters no errors, the API sends a message to the JD Edwards EnterpriseOne Work Center indicating that the job completed successfully.

Prerequisites

Before you begin setting up batch error messages, ensure that you:

- Create a batch application object.
- Complete the design of the report template with considerations for defining batch error messages.

Creating Data Items for Level Break Messages

Access Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW).

1. Add a data item to the project in which you want to include the level break message.
2. On the Data Dictionary Item Type form, click Yes to create a glossary data item.

3. On the Glossary Items form, enter the alias of the level break message in the Alias field.
4. Click the visual assist in the Glossary Group field and select glossary group Y.
5. In the Product Code field, enter a product code in the 55–59 range, which is reserved for clients.
6. In the Product Code Reporting field, enter a product code that represents the product that uses the level break message.
7. In the Description field, enter a meaningful description.
If this is a text-substituted message, enter the description using ampersands and sequential numbers to hold positions for substituted variables.
8. Click the visual assist in the Error Level field and select 2.
9. Select the Item Glossary tab and enter a cause and resolution in the text area.
10. Save the glossary data item.

Creating Business Function Data Structures

Access the appropriate project in OMW.

1. Add a data structure to the same project in which you added the other level break message components.
2. Select the Design Tools tab and click Data Structure Design.
3. On the Data Structure: Level break message form, enter the alias of the required data item on the Dictionary Items Alias field.
4. Double-click the data item to include it in the data structure.
You can also drag the data item to Structure Members.
5. Enter *EV01* in the Alias field and include it in the data structure.
6. Change the structure member name of the *cEverestEventPoint01* data item to *cIncludeInterconnect*.
7. Enter *GENLNG* in the Alias field and include it in the data structure.
8. Change the structure member name of the *idGenericLong* data item to *idGenlong*.

Creating Level Break Message Business Functions

Access the appropriate project in OMW.

1. Add a business function to the same project in which you added the other level break message components.
2. Select the Design Tools tab and click Start Business Function Design Aid.
3. On the Business Function Design form, click the visual assist in the Parent DLL field and select the DLL that your company uses for creating custom business functions.
4. In the detail area, enter a unique name in the Function Name field.
This name cannot include any spaces and should be descriptive of the business function purpose.
5. Enter a description in the Description field.
This description can be the same as the name, however, you can use spaces in the description.
6. Tab through the remaining fields.
The system populates the F3 Code field with a 3 (Minor Business Rule).

7. Modify the F3 Code if appropriate.
8. Click the row header to highlight the row of data that you just entered, and select Parameters from the Row menu.
9. Use the QBE (query by example) line to locate the associated data structure, click the data structure, and then click Select.

This process attaches the business function data structure that was created for the level break message to the business function.

10. On the Business Function Design form, verify that the name of the appropriate data structure appears in the Template Name field in the detail area.
11. From the Form menu, select Create to begin the creation of the business function.
12. Click Yes to answer the question Functions Not Found: Would you like skeletons created?
13. Click Yes to answer the question Function Prototypes Not Found: Would you like them to be created? and click OK.

The last message that appears reminds you to create a TYPEDEF for the data structure.

14. On the Business Function Design form, click the row header to highlight the row and select Typedef from the Row menu.

On the status bar, the message Your typedef is in the clipboard appears. This process creates the TYPEDEF for the data structure attached to the business function.

15. From the Form menu, select Edit.

The system launches Microsoft Visual C ++.

16. From the Window menu, select to work with the .h file.

You can also select the .h file using the tabs.

17. Locate the DS Template Type Definitions section of the .h file and just below the heading, paste the TYPEDEF using CTRL V.

This process copies the TYPEDEF for the data structure that is attached to the business function.

18. Save the business function and minimize Microsoft Visual C ++.

19. On the Business Function Design form, save the business function.

If you are not using a test substituted error message, skip to step 22.

20. Return to OMW and locate the level break message data structure for the text substitution, enter design, select the Design Tools tab, and click Create a type definition.

On the status bar, the message Your typedef is in the clipboard appears.

21. Return to Microsoft Visual C ++, locate the Structure Definitions section of the .h file and just below the heading, paste the TYPEDEF using CTRL V.

This process copies the TYPEDEF for the text substitution data structure into the business function.

22. Return to OMW and add the associated application to the project, enter design, select the Design Tools tab, click Start Form Design Aid, select the form to be launched, and select Application Properties from the File menu.

If a link to an associated application is not required in the message, skip to step 25.

23. On the Application Properties form, select the Operations tab, click Generate Form Data Structures, and in Notepad, highlight the entire form data structure for the appropriate form and copy the section.

24. Return to Microsoft Visual C ++, locate the Structure Definitions section of the .h file and just below the heading, paste the TYPEDEF.

This TYPEDEF resides above the TYPEDEF that was pasted from the level break message data structure for the text substitution.

25. Locate the External Business Function Header Inclusions section of the .h file and enter a call to the business function to process the message to Work Center.

Begin the call with #include (for example #include<B0100011.h >).

26. From the Window menu, select the .c file.

You can also select the .c file using the tabs.

27. Enter the appropriate code in the .c file as described in the Sample Source Code section.

28. When the business function is complete, save the business function, and click Build Business Function from the Design Tools tab.

Calling the Work Center Initialization API

Open a report in which you want to add batch error messaging in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

1. Create a report scope event rule variable in Event Rules Design using the data dictionary item *GENLNG*.
2. Access the appropriate event and select to call a business function.

Generally, this business function is called in Event Rules Design from the primary section of the report using the Initialize Section event.

3. Use the QBE line to select the JD Edwards EnterpriseOne Work Center initialization business function *B0100025* with a description of F01131 Edit JDEM Error Message.
4. Refer to the Windows Help file of APIs to identify the appropriate parameters to pass.

Calling the Processing Work Center APIs

Access RDA for a report in which you want to add batch error messaging.

1. Define all appropriate level breaks for the report.

You need to analyze the events that logically group all errors. This typically happens at events in which all editing has been completed for a group of records or immediately after all editing for an individual record has occurred.

2. For each level break established, call the business function for the level break message at the appropriate level break.

The business function for the level break message should relate to the type of error grouping that you want to capture at the particular level break. For example, SetLevel_SFVoucher groups errors that are related at the voucher level break. For reports, this business function is typically called in the Do Section event. If the interconnect is blank, then the business function is not calling an action message.

3. Call the JD Edwards EnterpriseOne Work Center error message business function immediately after the call to the level break message.

This name of this business function is *B0100011* and the description is Process batch errors to JDEM system.

4. Refer to the Windows Help file of APIs to identify the appropriate parameters to pass.

Glossary of JD Edwards EnterpriseOne Terms

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 relational databases, flat file format, and other data or message encoding, such as XML.
alternate currency	<p>A currency that is different from the domestic currency (when dealing with a domestic-only transaction) or the domestic and foreign currency of a transaction.</p> <p>In JD Edwards EnterpriseOne Financial Management, alternate currency processing enables you to enter receipts and payments in a currency other than the one in which they were issued.</p>
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).
as if processing	A process that enables you to view currency amounts as if they were entered in a currency different from the domestic and foreign currency of the transaction.
as of processing	A process that is run as of a specific point in time to summarize transactions up to that date. For example, you can run various JD Edwards EnterpriseOne reports as of a specific date to determine balances and amounts of accounts, units, and so on as of that date.
Auto Commit Transaction	A database connection through which all database operations are immediately written to the database.
back-to-back process	A process in JD Edwards EnterpriseOne Supply Management that contains the same keys that are used in another process.
batch processing	<p>A process of transferring records from a third-party system to JD Edwards EnterpriseOne.</p> <p>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.</p>
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 immediate	<p>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.</p> <p>See also direct connect and store-and-forward.</p>
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 categorization	A way to categorize business service properties. These properties are categorized by business service.
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).
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.
contra/clearing account	A general ledger account in JD Edwards EnterpriseOne Financial Management that is used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations in JD Edwards EnterpriseOne Financial Management.
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.
cost assignment	The process in JD Edwards EnterpriseOne Advanced Cost Accounting of tracing or allocating resources to activities or cost objects.
cost component	In JD Edwards EnterpriseOne Manufacturing, an element of an item's cost (for example, material, labor, or overhead).
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.
cross segment edit	A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced.
currency restatement	The process of converting amounts from one currency into another currency, generally for reporting purposes. You can use the currency restatement process, for example, when many currencies must be restated into a single currency for consolidated reporting.
cXML	A protocol used to facilitate communication between business documents and procurement applications, and between e-commerce hubs and suppliers.
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.
date pattern	A calendar that represents the beginning date for the fiscal year and the ending date for each period in that year in standard and 52-period accounting.

denominated-in currency	The company currency in which financial reports are based.
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.
detail information	Information that relates to individual lines in JD Edwards EnterpriseOne transactions (for example, voucher pay items and sales order detail lines).
direct connect	A transaction method in which a client application communicates interactively and directly with a server application. See also batch-of-one immediate and store-and-forward.
Do Not Translate (DNT)	A type of data source that must exist on the iSeries because of BLOB restrictions.
dual pricing	The process of providing prices for goods and services in two currencies.
duplicate published business services authorization records	Two published business services authorization records with the same user identification information and published business services identification information.
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 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 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 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.
facility	An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. A facility is sometimes referred to as a “business unit.”
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.
header information	Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows.
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.
IServer service	This internet server service resides on the web server and is used to speed up delivery of the Java class files from the database to the client.
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.
local standalone BPEL/ESB server	A standalone BPEL/ESB server that is not installed within an application server.
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.
matching document	A document associated with an original document to complete or change a transaction. For example, in JD Edwards EnterpriseOne Financial Management, a receipt is the matching document of an invoice, and a payment is the matching document of a voucher.
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.
Middle-Tier BPEL/ESB Server	A BPEL/ESB server that is installed within an application server.
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 than 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.
<i>nota fiscal</i>	In Brazil, a legal document that must accompany all commercial transactions for tax purposes and that must contain information required by tax regulations.
<i>nota fiscal factura</i>	In Brazil, a <i>nota fiscal</i> with invoice information. See also <i>nota fiscal</i> .
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-in capabilities 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).
planning family	A means of grouping end items whose similarity of design and manufacture facilitates being planned in aggregate.
preference profile	The ability to define default values for specified fields for a user-defined hierarchy of items, item groups, customers, and customer groups.
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-grade file server	A file server that has been quality assurance tested and commercialized and that is usually provided in conjunction with user support services.
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	<p>The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):</p> <p>11>21>26>28>38>01</p> <p>In this path, <i>11</i> equals new project pending review, <i>21</i> equals programming, <i>26</i> equals QA test/review, <i>28</i> equals QA test/review complete, <i>38</i> equals in production, <i>01</i> 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.</p>
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.
pull replication	One of the JD Edwards EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers using JD Edwards EnterpriseOne data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the F98DRPCN table.
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, such as B73.2 or B73.2.1.
replication server	A server that is responsible for replicating central objects to client machines.
Rt-Addressing	Unique data identifying a browser session that initiates the business services call request host/port user session.
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.
quote order	In JD Edwards Procurement and Subcontract Management, a request from a supplier for item and price information from which you can create a purchase order. In JD Edwards Sales Order Management, item and price information for a customer who has not yet committed to a sales order.
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.
SEI implementation	A Java class that implements the methods that declare in a Service Endpoint Interface (SEI).
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.
Service Endpoint Interface (SEI)	A Java interface that declares the methods that a client can invoke on the service.
SOA	Abbreviation for Service Oriented Architecture.
soft coding	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.
spot rate	An exchange rate entered at the transaction level. This rate overrides the exchange rate that is set up between two currencies.
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.
superclass	An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the superclass of "Oak" and "Elm," for example.
supplemental data	<p>Any type of information that is not maintained in a master file. Supplemental data is usually additional information about employees, applicants, requisitions, and jobs (such as an employee's job skills, degrees, or foreign languages spoken). You can track virtually any type of information that your organization needs.</p> <p>For example, in addition to the data in the standard master tables (the Address Book Master, Customer Master, and Supplier Master tables), you can maintain other kinds of data in separate, generic databases. These generic databases enable a standard approach to entering and maintaining supplemental data across JD Edwards EnterpriseOne systems.</p>
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.
three-tier processing	The task of entering, reviewing and approving, and posting batches of transactions in JD Edwards EnterpriseOne.
three-way voucher match	In JD Edwards Procurement and Subcontract Management, the process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records to create vouchers.
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.
two-way authentication	An authentication mechanism in which both client and server authenticate themselves by providing the SSL certificates to each other.
two-way voucher match	In JD Edwards Procurement and Subcontract Management, the process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information.
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.
variance	<p>In JD Edwards Capital Asset Management, the difference between revenue generated by a piece of equipment and costs incurred by the equipment.</p> <p>In JD Edwards EnterpriseOne Project Costing and JD Edwards EnterpriseOne Manufacturing, the difference between two methods of costing the same item (for example, the difference between the frozen standard cost and the current cost is an engineering variance). Frozen standard costs come from the Cost Components table, and the current costs are calculated using the current bill of material, routing, and overhead rates.</p>

versioning a published business service	Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.
Version 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.
wchar_t	An internal type of a wide character. It is used for writing portable programs for international markets.
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 proxy foundation	Foundation classes for web service proxy that must be included in a business service server artifact for web service consumption on WAS.
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.
work day calendar	In JD Edwards EnterpriseOne Manufacturing, a calendar that is used in planning functions that consecutively lists only working days so that component and work order scheduling can be done based on the actual number of work days available. A work

	day calendar is sometimes referred to as planning calendar, manufacturing calendar, or shop floor calendar.
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

Numerics/Symbols

52 Period Column Headings
(P83110B) 237

A

absolute position, activating 101
accounting periods, assigning to column
headings 238
Acrobat Reader form 158
action messages
 defining in business functions 282
 definition 276
 understanding 277
Add Argument List form 263
Add EnterpriseOne Object to the project
form 23
Add Object form 23
additional documentation xx
Additional Properties form 57
additional properties, defining for
application reports 57
Adobe Acrobat Reader, viewing
reports 58
Advance Section event
 description 201
 values held in the runtime
 structures 208, 210, 214
Advanced Operations form 178
After Last Object Printed event
 description 201
 values held in the runtime
 structures 209, 213
aggregates
 adding to an exiting level break
 footer 136
 descriptions 134
 selecting operators and operands 136
Aggregations form 133, 136
Align Objects form 99
alignment
 aligning fields within or across
 sections 100
 justifying text 117
 setting the grid alignment 66
Alignment Grid form 66

alpha variables
 description 79
alternative descriptions
 setting up for business view
 favorites 226
API
 adding records to the subsystem
 table 247, 248
 definition 162
application fundamentals xix
Application Programming Interface, *See*
API
application reports
 creating batch versions of 57
 creating calculation columns in 55
 defining additional properties 57
 defining data sequencing in 56
 defining level breaks 56
 defining page breaks 56
 defining sort properties 56
 description 34
 format 8
 selecting business views in 54
 selecting records to include in 56
 selecting Report Director templates 54
 selecting smart fields in 54
 understanding 51
Arabic 25
Assignment form 170, 171, 173
assignments
 creating in event rules 170
 understanding 168
 using in level break footers 134
Associated Description Properties
form 136
associated descriptions
 adding to level break headers 135
 understanding 133
asynchronous
 defining for report interconnects 218
 processing subsystem jobs 245
asynchronous processing 217
attachments, adding and deleting 124
audit trails, reviewing 158
automatically generated rows 88, 92

B

- bar code fonts, setting up 113
- Batch Application Design form 23
- batch error messages
 - calling Work Center APIs 277
 - first-level messages 277
 - second-level messages 277
 - third-level messages 277
 - understanding 275
 - understanding components 278
 - using action messages 277
 - using level break messages 276
 - using text substituted messages 277
- batch events
 - Do Section 200
 - report level events 200
 - section level events 201
 - understanding 200
 - using system functions 204
- batch processes
 - creating as subsystem jobs 245
 - creating in Report Design Aid 188
- batch processing
 - processing batch events 200
 - processing data selection and data sequencing 195
 - processing events and attached event rules 207
 - processing level break sections 197
 - processing report sections 188
 - processing sections 190
 - understanding 188
 - using database output to update the database 161
- batch runtime processing
 - objects available 207
 - understanding 207
- batch versions
 - copying XJDE and ZJDE versions 22
 - creating from the Report Director 37, 43, 48, 57
 - deleting from multiple locations 22
 - opening 22
 - overview 14
- Before Level Break event
 - description 201
 - values held in the runtime structures 209, 211
- boxes
 - associating with data fields 121
 - understanding 121
- BrowsER, *See* JD Edwards EnterpriseOne BrowsER
- Browsing form 179
- Business Function Design form 248
- Business Function Search form 236
- business functions
 - creating for level break messages 280
 - understanding 52
- Business Functions form 236
- business view columns
 - adding and removing in detail sections 26
 - selecting for columnar sections 36
 - selecting for group sections 41
 - selecting for tabular sections 46
 - understanding 77
- Business View Columns Browser, showing and hiding 66
- business view favorites
 - adding business views 224
 - adding notes 225
 - adding to favorites folders 222
 - adding to favorites subfolders 223
 - modifying and deleting notes 226
 - selecting 30, 54
 - setting up 221
 - setting up alternative descriptions 226
 - understanding 221
 - viewing alternative descriptions 228
- Business View Selection Option form 36
- business views
 - adding to favorites folders and subfolders 224
 - selecting for columnar sections 36
 - selecting for detail sections 26, 29
 - selecting for group sections 41
 - selecting for tabular sections 46
 - selecting from favorites 30, 54
 - selecting from the Favorite Business Views tab 30
 - selecting from the Select Business View tab 29
 - setting up favorites 221
 - understanding business view columns 77
 - using in subsection joins 147

C

- calculation columns
 - creating 86
 - creating in tabular sections 55
 - defining 87
 - removing calculations 88
 - selecting calculation types 53
 - understanding 53
- Calculation Row Properties form 90
- calculation rows
 - adding 90
 - understanding 88
- Cell Properties form 94, 95
- cells
 - defining 93
 - defining in tabular sections 83
 - overriding constant row variables 94
 - overriding data row variables 94
 - overriding properties 93
 - overriding underline row variables 94
- child sections
 - available events 182
 - understanding 147
- colors
 - for fonts 110
 - supported for printing 110
- Column Data Selection form 55
- Column Heading Properties form 75
- column headings
 - assigning account periods 238
 - changing text 75
 - in columnar sections 9
 - naming smart field columns 55
 - renaming the Row Description column 51
- Column Headings (P83110) 237
- Column Inclusion event
 - comparing to the Do Section event 175
 - using 176
- column properties
 - modifying from the Column tab 75
 - understanding 69
 - viewing and modifying 74
- column spacing, changing 102
- Column Variable Properties form 85
- column width, changing 108
- columnar section
 - changing the length of fields 108
- Columnar Section form 71
- columnar section report
 - creating an example 38
- columnar section reports
 - creating batch versions of 37
 - including sections in 35
- Columnar Section Spacing form 103
- columnar sections
 - adding and removing business view columns 26, 77
 - adding and removing data fields 80
 - available events 182
 - changing the width of columns 108
 - characteristics of 9
 - creating level break headers 135
 - creating with the Report Director 35
 - deciding when to use 9
 - defining 35
 - defining as conditional 9
 - defining data selection 27
 - defining data sequencing 27
 - defining data sequencing in 37
 - defining sort order 27
 - defining sort properties 27
 - defining sort properties in 37
 - displaying conditionally 98
 - example 9
 - filtering data 27
 - hiding 98
 - modifying row spacing 103
 - processing 191
 - renaming 97
 - reprinting last line on next page 104
 - selecting business view columns in 36
 - selecting business views 26
 - selecting business views in 36
 - selecting records to include in 37
 - setting up totals 9
 - understanding 35
 - using in subsection joins 148
- columns
 - adding and removing business view columns 26
 - aligning within or across sections 99, 100
 - changing spacing 102
 - creating calculations 86
 - defining in tabular sections 83
 - modifying column width 107
 - removing from section layout 26, 27
 - understanding smart fields 139

- using Row Description columns 84
- comments
 - adding to reports 123
 - adding, modifying, and deleting on a data field 125
- comments, submitting xxiv
- common fields xxiv
- company title
 - description 79
 - See Also* runtime fields
- comparison, defining for data selection 27
- conditional sections
 - defining columnar sections as 9
 - defining group sections as 10
- constant events 203
- Constant Row Properties form 91
- constant rows
 - adding 91
 - overriding variable properties 94
 - overriding variables 94
- constants
 - description 79
- contact information xxiv
- Copy Object form 24
- copying
 - report templates in OMW 24
- Create New Report form 23
- Criteria Design form 170
- cross-references xxiii
- CSV
 - setting the grid alignment 65
 - showing the Tip Dialog box 64
- custom sections
 - attaching logic 177, 195
 - available events 182
 - creating 176
 - understanding 176, 195
- Customer Connection website xx

D

- Data Dictionary Browser form 242
- Data Dictionary Browser, showing and hiding 67
- data dictionary fields
 - creating for level break messages 278
 - showing and hiding text overrides 66
 - understanding 78
 - using to create event rule variables 174
- Data Dictionary Item Type form 261, 264
- data fields
 - adding and removing 80
 - aligning within or across sections 99
 - associating boxes and lines 121
 - changing colors 112
 - changing decimal scaling for 85
 - changing font properties 112
 - changing names of 81
 - inserting into columnar sections 80
 - inserting into page footers 129
 - inserting into page headers 129
 - inserting into report footers 129
 - inserting into report headers 128
 - justifying text (alignment) 117
 - modifying field length 107
 - types of 79
 - understanding 78
- Data Item Specifications form 261, 264
- Data Row Properties form 90
- data rows
 - adding 90
 - overriding variable properties 94
 - overriding variables 94
 - understanding 88
- data selection
 - defining in columnar, group, and tabular sections 27
 - locking 29
 - processing of 195
- Data Selection form 32
- data sequencing
 - defining in columnar sections 27, 37
 - defining in group sections 27, 42
 - defining in subsection joins 149
 - defining in tabular sections 27, 47
 - defining using the Advanced Option 56
 - processing of 195
- Data Sequencing form 31
- Data Sequencing Help form 51
- Data Structure:Level break message form 285
- data structures
 - creating for level break message business functions 279
 - creating for text substituted level break messages 279
 - creating processing option data structures 241
- data, filtering in columnar, group, and tabular sections 27
- database maintenance

- using database output 161
 - using table conversions 161
 - using table I/O (in event rules) 161
 - database output
 - description 162
 - updating the database using batch applications 161
 - updating, inserting, and deleting records 161
 - working with 161
 - Date Title Preview form 234
 - Date Title Revisions form 233
 - Date Title Search form 236
 - date titles
 - adding to financial reports 236
 - assigning accounting periods to column headings 237
 - customizing 233
 - defining custom 233
 - previewing 234
 - shipped with the software 231
 - using in financial reports 232
 - Date Titles (P83100) 232
 - date variable data fields 80
 - date variables
 - description 79
 - decimal scaling
 - changing for all fields in a report 86
 - changing for all fields in a section 86
 - changing for individual fields 85
 - definition 85
 - default report settings 20
 - Define Calculation form 56
 - Define Sort Properties form 37, 42, 47
 - Define Sub Section Join form 149
 - Delete of form 24
 - deleting, report objects 24
 - dependent sections, *See* report sections
 - description columns, *See* Row Description columns
 - descriptions, associating in level break headers 135
 - detail sections
 - adding and removing business view columns 26, 77
 - adding columns using Quick Section 30
 - adding from the menu 26, 29
 - adding in Report Design Aid 26
 - definition 8
 - design capabilities 8
 - displaying conditionally 98
 - formats 7
 - hiding 98
 - joining 149
 - renaming 97
 - setting page breaks 105
 - types 25
 - understanding 25
 - Director, *See* Report Director
 - Director templates, *See* Report Director templates
 - Director Templates (P91400) 51
 - Director's Finish form 37
 - display tree
 - showing and hiding 66
 - understanding 65
 - Do Balance Auditor event, description 201
 - Do Initialize Printer event, description 200
 - Do Section event
 - description 201
 - understanding 200
 - values held in the runtime structures 209, 212
 - Do Tabular Break event, description 201
 - documentation
 - printed xx
 - related xx
 - updates xx
 - double byte fonts
 - aligning 118
 - understanding 111
 - drill down
 - activating 155
 - defining 156, 157
 - description 145
 - understanding 155
 - dynamic positioning
 - activating 114
 - applying font substitutions to report templates 116
 - defining font substitutions 114
 - overriding font substitutions 116
 - understanding 114
- E**
- edit codes
 - selecting 119
 - understanding 119

- End Break Section event, description 201
- End Lvl Brk Footer Section event 199
 - description 201
- End Lvl Brk Header Section event 199
 - description 201
- End Report event, description 200
- End Section event, description 201
- error messages, *See* batch error messages
- event flow
 - processing the Advance Section
 - event 208, 210, 214
 - processing the After Last Object Printed
 - event 209, 213
 - processing the Before Level Break
 - event 209, 211
 - processing the Do Section event 212
 - processing the Initialize Section
 - event 208
- event levels
 - object events for constants 184
 - object level events for variables 184
 - report level events 182
 - section level 182
 - understanding 182
- event rule assignments, creating 171
- event rule variables
 - creating 175
 - naming 174
 - understanding 174
- event rules
 - calling system functions in 172
 - changing sequence 170
 - creating assignments 170
 - creating event rule variables 174
 - creating for processing options 181
 - creating if/while statements 167
 - selecting from available fields 168
 - understanding 167
 - understanding events 181
 - using Column Inclusion or Do
 - Section 175
 - using text variables in 171
- Event Rules Design form 170, 218
- Event Rules Variables form 175
- events
 - constants and variables 203
 - object level events for constants 184
 - object level events for variables 184
 - page footer section 202
 - page header section 202

- report footer section 203
- report header section 202
- report level 182
- section level 182, 201
- understanding 181
- Expression Manager form 88, 90
- expressions, understanding 168

F

- favorite business views, *See* business view
- favorites
 - selecting from the Favorite Business
 - Views tab 30
- Favorites (P9100) 221
- field length, changing 108
- field properties, viewing 69
- fields
 - aligning within or across sections 99, 100
 - changing font properties 112
 - descriptions of prefix codes 168
 - justifying text 117
 - modifying length 107
 - selecting for detail section layout 26
- Fields property tab 71
- Fields tab 74, 76
- Financial Account Level of Detail Row
 - Generation form 92
- financial report date titles 232
- first-level messages, understanding 277
 - See Also* level break messages
- Font form 117
- font properties
 - changing for all fields in a report 112
 - changing for all fields in a section 112
 - changing for an individual field 112
- Font Substitution by Language Revisions
 - form 115
- Font Substitution by Language Type
 - Revisions form 116
- font substitutions
 - applying to report templates 116
 - defining 114
 - defining for language and line
 - printers 115
 - modifying 116
 - overriding 116
 - viewing by language type 115
- fonts
 - activating dynamic positioning 114

- aligning bar codes 118
- aligning double byte 118
- assigning by language 117
- changing properties 109
- changing when printing 114
- selecting colors 110
- setting up bar codes 113
- understanding barcodes 112
- understanding defaults 20
- understanding double byte 111
- understanding proportional and nonproportional 109
- using True Type 116
- footer sections
 - characteristics of page footers 13
 - characteristics of report footers 14
 - creating 128
 - defining level break fields 8
 - understanding page footers 7
 - understanding report footers 7
- Form Interconnections form 156, 158

G

- general, description of system
 - functions 204
- Glossary Items form 285
- grid alignment
 - setting 66
 - understanding 65
- Group Section form 71
- group section reports
 - creating an example 43
 - creating batch versions of 43
 - including sections in 41
- group sections
 - adding and removing business view columns 26, 77
 - adding using Report Design Aid 26
 - available events 182
 - characteristics 9
 - creating level break headers 135
 - creating with the Report Director 40
 - deciding when to use 10
 - defining 40
 - defining as conditional 10
 - defining data selection 27
 - defining data sequencing 27
 - defining data sequencing in 42
 - defining sort order 27
 - defining sort properties 27

- defining sort properties in 42
- displaying conditionally 98
- example 9
- filtering data 27
- hiding 98
- organization of fields 40
- processing 191
- renaming 97
- reprinting last line on next page 104
- selecting business view columns in 41
- selecting business views 26
- selecting records to include in 42
- setting up totals 10
- typical event flow 208
- understanding 39
- using in subsection joins 148
- groups sections
 - selecting business views in 41

H

- header sections
 - characteristics of 13
 - creating 127
 - defining level break fields 8
 - populating page headers
 - automatically 35
 - understanding page headers 7
 - understanding report headers 7
- Help with Section Data Selection form 56

I

- if/while statements
 - creating in event rules 170
 - understanding 168
- implementation guides
 - ordering xx
- independent sections, *See* report sections
- Init Break Section event, description 201
- Init Lvl Brk Footer Section event
 - understanding 199
- Init Lvl Brk Footer Section event, description 201
- Init Lvl Brk Header Section event
 - understanding 199
- Init Lvl Brk Header Section event, description 201
- Initialize Report event, description 200
- Initialize Section

- values held in the runtime
 - structures 208
- Initialize Section event
 - description 201
- invisible sections
 - defining in properties 99
 - showing and hiding 66
- item properties 74
 - See Also* field properties

J

- JD Edwards EnterpriseOne BrowsER
 - accessing for a report template 179
 - accessing for report templates 178
 - understanding 178
- JD Edwards Report Design Aid form 23
- JDE.DataItem Properties form 242
- Job Submission form 113
- joined sections
 - attaching logic 195
 - modifying and severing 149
 - processing 195
 - understanding the join 148

L

- Language Font Revisions form 117
- language, setting up alternative descriptions
 - for business view favorites 226
- left operand, defining for data selection 27
- level break fields
 - defining 135
 - hiding in the detail section 135
 - in application reports 51
 - in tabular sections 45, 47
- Level Break Footer form 136
- level break footers
 - available events 182
 - creating 136
 - defining for group and columnar
 - sections 136
 - defining level break fields 8
 - inserting descriptions 136
 - modifying 134
 - reprinting at page break 136
 - selecting aggregates 136
 - selecting operators 133
 - understanding 133
 - using assignments in 134
- Level Break form 135, 136

- level break headers
 - associating descriptions 135
 - available events 182
 - creating 135
 - defining for group and columnar
 - sections 135
 - defining level break fields 8
 - modifying 134
 - understanding 131, 132
- level break messages
 - creating business function data structures
 - for 279
 - creating business functions for 280
 - creating data items for 278
 - creating data structures for 279
 - defining level of messages 277
 - understanding 276
 - understanding components 278
 - using action messages 277
- level break processing
 - End Lvl Brk Header Section and End Lvl
 - Brk Footer Section events 199
 - understanding 197
- level break section events
 - End Lvl Brk Header Section and End Lvl
 - Brk Footer Section 199
 - Init Lvl Brk Footer Section 199
 - Init Lvl Brk Header Section 199
 - understanding 199
- level break sections
 - defining for columnar sections 9
 - defining for group sections 10
 - Init Lvl Brk Footer Section event 199
 - Init Lvl Brk Header Section 199
 - processing 197
 - processing events 199
- level breaks fields
 - defining in application reports 56
- level breaks, understanding 8
- levels, understanding 131
- lines
 - associating with data fields 121
 - understanding 121
- List of values form 29
- locking data selection 29
- logic 8
 - See Also* event rules
 - creating for custom sections 177, 195
 - creating in joined sections 195

understanding processing option
logic 181

M

Mapping Targets form 163
Mappings form 163
margins, showing and hiding 66
master specifications
 overriding in batch versions 15
 reading by batch versions 14
 understanding 14
media objects, description of system
 functions 204
messages
 action messages 277
 batch error messages 275
 level break messages 276
 text substituted messages 277
 understanding level break
 messages 276
messaging
 batch error messages 275
 description of system functions 204

N

Named Event Rules Design form 264
Named Mapping form 263
naming conventions
 event rule variables 174
 modifying batch versions beginning with
 XJDE and ZJDE 22
 naming report objects 19
 Report Director templates 252
Navigation Assistant
 showing and hiding 66
 understanding user options for 64
notes xxiii
 adding to favorites, folders, and
 subfolders 225
 modifying and deleting from favorites,
 folders, and subfolders 226
Notes Revisions form 225
number of non optimized inclusion
 rows 89
number of optimized inclusion rows 89
numeric fields
 changing decimal scaling for 85
 changing the appearance of 120
 understanding formatting 118

numeric variable data fields 80
numeric variables
 description 79
 formatting 118

O

Object Design Properties form 75
Object Folder Revisions form 222
object level events
 descriptions of constant and variable
 events 203
 events available for constants 184
 events available for variables 184
Object Management Workbench
 copying report templates 24
 creating report objects 23
 opening report objects 24
Object Management Workbench form 23
objects
 batch runtime processing 207
 creating in Object Management
 Workbench 23
 description of system functions 204
 naming 19
 understanding in JD Edwards
 EnterpriseOne 7
 See Also report objects
 working with 77
operator, defining for data selection 27
optimize rows, understanding 89

P

page breaks
 defining in application reports 56
 inserting manually 105
 understanding manual page breaks 105
page footer events 202
Page Footer form 129
page footers
 available events 182
 characteristics of 13
 creating 129
 description 128
 example 14
 fields used in 13
 understanding 127
 using data fields in 129
Page Header Details form 35, 41, 46, 54

- page header events, descriptions of each 202
- Page Header form 129
- page headers
 - adding data fields 78
 - available events 182
 - characteristics 13
 - creating 129
 - description 128
 - example 13
 - fields used in 13
 - populating automatically 35
 - understanding 127
 - using data fields 129
- Page n of Total, description 79
 - See Also* runtime fields
- page number data fields 80
- page number, description 79
 - See Also* runtime fields
- parent sections
 - understanding 147
- parent/child relationships, subsection joins 147
- PeopleCode, typographical conventions xxii
- percent calculations, defining 88
- performance enhancement in tabular sections, row optimization 89
- prefix codes, descriptions 168
- prerequisites xix
- preview
 - changing number of lines to display 66
 - previewing date titles 234
 - previewing reports from Report Design Aid 58
- printed documentation xx
- printing
 - colors supported 110
 - printing text attachments on reports 152
 - repeating the last line of a page as the first line on the succeeding page 104
- Processing Option Design form 241
- processing option logic
 - adding to the report 243
 - understanding 181
- processing option template
 - adding logic to the report 243
- processing option templates
 - adding tabs 242
 - attaching to report templates 243

- creating 241
- designing 240
- guidelines for basic event rules 181
- understanding 239
- properties
 - changing fonts 109
 - changing fort properties 112
 - defining sort properties 27
 - modifying column properties 74
 - modifying field properties 74
 - modifying row properties 76
 - overriding cell properties 93
 - understanding report object properties 97
 - understanding report properties 70
 - understanding section properties 71
- Properties form 70, 248
- Purge Financial Reporting Drill Down Work File form 158

Q

- Quick Section form 26
- Quick Section, using to add business view columns to detail sections 30

R

- Range of values form 29
- records
 - selecting to include in columnar sections 37
 - selecting to include in group sections 42
 - selecting to include in tabular sections 47
- Refresh Section event, description 202
- related documentation xx
- report components
 - default standards for 20
 - understanding 6
- Report Data Structure form 248
- report date, description 79
 - See Also* runtime fields
- report design
 - adding and removing attachments and comments 123
 - creating report models 18
 - determining report sections 18
 - guidelines 83
 - reviewing 57

- surveying stakeholders 17
 - understanding the process 17
- Report Design Aid
 - configuring the design workspace 63
 - creating batch processes 188
 - understanding 6
 - understanding the design workspace 63
- Report Director
 - creating application reports 51
 - creating columnar sections 35
 - creating detail sections using 26
 - creating group section reports 40
 - creating tabular sections 45
 - customizing directors 6
 - understanding 33
- Report Director forms
 - Column Data Selection 55
 - Define Calculation 56
 - Define Sort Properties 37, 42, 47
 - defining Smart Field Parameters 55
 - Page Header Details 54
 - Section Data Sequencing 56
 - Section Layout 36, 41, 46
 - Smart Field Data Selection 55
 - understanding 6
- Report Director templates
 - adding 253
 - adding and modifying 251
 - additional properties 51
 - defining default criteria 33
 - selecting 54
 - understanding 34, 251
- Report Director Templates Revisions
 - form 252
- report footer events 203
- Report Footer form 130
- report footers
 - available events 182
 - characteristics of 14
 - creating 129
 - description 128
 - example 14
 - fields used in 14
 - understanding 127
 - using data fields 129
- report formats 7
- report header events 202
- Report Header form 129
- report headers
 - available events 182
 - characteristics of 12
 - creating 128
 - description 128
 - example 12
 - fields used in 12
 - understanding 127
 - using data fields 128
- report interconnects
 - asynchronous and synchronous processing 217
 - creating 218
 - in batch applications 217
 - understanding 217
- report level events
 - descriptions of each 200
 - events available 182
- report model, example 18
- report objects
 - available for batch processing 207
 - business view columns 77
 - creating from Object Management Workbench 23
 - creating from Report Design Aid 23
 - data fields 77
 - definition 7
 - deleting 24
 - deleting from multiple locations 22
 - description of alphabetical codes 207
 - descriptions of each type 79
 - information required for 21
 - opening from Object Management Workbench 24
 - understanding 21
- report planning checklist 19
- Report Preview form 58
- report processing
 - asynchronous and synchronous report interconnects 217
 - optimizing tabular row processing 89
 - overview 6
 - processing events 200
 - processing level breaks 197
 - processing sections 188, 190
 - understanding 187
- report sections
 - adding and removing business view columns 80
 - adding lines and boxes 122
 - aligning fields across 101
 - aligning fields within 100

- changing decimal scaling in tabular sections 86
- changing descriptions 98
- changing font properties 112
- characteristics of columnar sections 9
- characteristics of group sections 9
- characteristics of page footers 13
- characteristics of page headers 13
- characteristics of report footers 14
- characteristics of report headers 12
- characteristics of tabular sections 10
- creating logic for custom sections 177
- creating totals for columnar and group sections 81
- custom sections 176, 195
- definition 7
- dependent (level-two) sections 188
- determining sections required 18
- hiding unconditionally 98
- independent (level-one) sections 188
- inserting additional sections 6
- joining 147
- joining sections in a report 145
- modifying column spacing 103
- parent sections 188
- processing 188
- selecting smart fields 54
- showing and hiding 98
- showing and hiding titles 66
- showing and hiding unconditionally 99
- understanding properties 71
- viewing event rules 178
- report templates
 - accessing JD Edwards EnterpriseOne BrowsER 178
 - applying font substitution 116
 - changing decimal scaling 86
 - changing font properties 112
 - copying from Object Management Workbench 24
 - default settings 20
 - deleting 24
 - deleting from multiple locations 22
 - opening 22
 - overview 14
 - reviewing event rules 179
 - reviewing results 57
 - saving 57
 - size guidelines 8
 - report time, description 79
 - See Also* runtime fields
 - report title, description 79
 - See Also* runtime fields
 - report writing, understanding 5
 - reports
 - overview 7
 - section processing 190
 - reports, understanding default settings 20
 - See Also* report templates
 - reviewing
 - report design 57
 - reports 58
 - right margin, showing and hiding 66
 - right operand, defining for data selection 28
 - Row Description columns
 - creating 84
 - deleting 84
 - including in application reports 51
 - including in tabular sections 11
 - working with 84
 - row optimization 89
 - row properties
 - modifying from section properties 76
 - viewing 69
 - viewing and modifying 76
 - row spacing
 - changing in columnar sections 103
 - changing in tabular sections with added rows 104
 - rows
 - adding calculation rows 90
 - adding constant rows 91
 - adding data rows 90
 - adding manually 90
 - adding sum rows 91
 - adding underline rows 91
 - changing spacing 103
 - defining in tabular sections 83
 - generating automatically 92
 - modifying spacing 104
 - rulers, showing and hiding 66
 - runtime engine 208
 - runtime fields
 - company title 79
 - Page n of Total, 79
 - page number 79
 - report date 79
 - report time 79
 - report title 79

runtime structures
 Do Section event 209
 values held after processing the Advance
 Section event 208, 210, 214
 values held after processing the After
 Last Object Printed event 209, 213
 values held after processing the Before
 Level Break event 209, 211
 values held after processing the Do
 Section event 212
 values held after processing the Initialize
 Section event 208

S

saving reports 57
 second-level messages,
 understanding 277
 See Also level break messages
 section data selection, defining 31
 See Also data selection
 section data sequencing, defining 31
 See Also data sequencing
 Section Layout form 36, 41, 46
 using in the Report Director 26
 section level events
 descriptions of each 201
 Do Section 200
 events available 182
 level break section 199
 page footer events 202
 page header events 202
 report footer events 203
 report header events 202
 section processing
 processing data selection and data
 sequencing 195
 processing group and columnar
 sections 191
 processing joined sections 195
 processing tabular sections 193
 understanding 190
 section properties, viewing 69
 section titles, showing and hiding 66
 sections
 description of system functions 204
 removing columns from the section
 layout 27
 removing columns from the Selected
 Columns list 26
 selecting for columnar section
 reports 35
 selecting for group section reports 41
 selecting for tabular section reports 46
 showing invisible sections 66
 understanding 7
 See Also report sections
 Select a Column form 263
 Select a Table form 263
 Select Business View form 30
 Select Columns form 54
 Select Processing Option Template
 form 243
 Select User Defined Code form 241
 Single value form 28
 Single Value form 236
 Smart Field Criteria form 261
 Smart Field Data Selection form 55
 Smart Field Name form 54
 Smart Field Parameters form 55
 Smart Field Template Revisions form 268
 smart field templates
 creating 268
 Smart Field Templates (P91420) 266
 smart fields
 adding to columnar and group
 sections 141
 adding to reports 139
 adding to tabular and application report
 sections 141
 creating custom 258
 creating templates 34
 selecting templates 140
 selecting to include in report
 sections 54
 understanding 52, 139, 257
 sort properties
 defining 31, 47
 defining for an application report 56
 defining in columnar sections 37
 defining in group sections 42
 understanding 27
 Sub Section Join form 148
 Submit Job-Submitted Job Search
 form 158
 Submit Job-Work With Batch
 Versions-Available Versions form 158
 subsection joins
 creating 147, 148
 defining data sequencing 149

- description 145
- joining existing detail sections 147
- modifying and severing 149
- understanding parent/child sections 147
- subsystem jobs
 - adding API records to the subsystem table 248
 - adding records to the subsystem table using API 247
 - defining 248
 - processing asynchronously 245
 - understanding 245
 - understanding definitions 247
- subsystem table, adding API records 247, 248
- suggestions, submitting xxiv
- Sum Row Properties form 91
- sum rows
 - adding 91
 - understanding 88
- Suspend Section event, description 202
- synchronous processing 217
- system functions
 - attaching to an event 172
 - descriptions of categories 204
 - understanding 168
 - using in batch events 204
 - using in event rules 173
- System Functions form 152, 173

T

- Tab Properties form 242
- table conversions
 - description 161
 - using to maintain the database 161
- Table I/O, description 162
- tabs
 - adding to processing option templates 242
 - showing and hiding in Report Design Aid 66
- tabular optimization 89
- tabular reports
 - creating with the Report Director 45
 - defining 45
- Tabular Section form 71
- tabular section reports
 - creating an example 48
 - creating batch versions of 48
 - including sections in 46
- tabular sections
 - activating drill down 155
 - adding and removing business view columns 26, 77
 - adding using Report Design Aid 26
 - advantages of using 12
 - automatic totaling 45
 - available events 182
 - changing decimal scaling 85, 86
 - characteristics 10
 - deciding when to use 12
 - defining calculation columns 87
 - defining data selection 27
 - defining data selection in 47
 - defining data sequencing 27
 - defining drill down 157
 - defining percent calculations 88
 - defining sort order 27
 - defining sort properties 27
 - defining sort properties in 47
 - design features available 11
 - design features not available 11
 - example 10
 - filtering data 27
 - hiding 98
 - including Row Description columns 11, 84
 - including totals 11
 - modifying column spacing 103
 - modifying row spacing 104
 - optimizing tabular row processing 89
 - overriding cells 94
 - overriding properties of cells 93
 - processing 193
 - renaming 97
 - reviewing audit trails 158
 - section properties 71
 - selecting business views 26
 - selecting business views columns in 46
 - selecting business views in 46
 - selecting records 47
 - selecting records to include in 47
 - understanding 45
 - understanding columns, rows, and cells 83
 - using spreadsheet functionality 45
 - working with objects 83
 - working with objects unique to tabular sections 83

- working with rows 88
- templates, *See* Report Director templates, report templates, smart field templates
- text
 - default settings 20
 - disconnecting from variable in group sections 81
 - justifying 117, 118
 - wrapping 101
- text attachments
 - adding to reports 151
 - description 145
 - printing on reports 152
- text justification
 - changing for variables 118
 - understanding 117
- text overrides, showing and hiding 66
- text substituted messages
 - creating business function data structures for 279
 - creating data items for 278
 - creating data structures for 279
 - understanding 277
- text variables
 - adding and using 171
 - creating 171
 - understanding 171
 - using in event rules 172
- Text Variables form 172
- text wrapping, activating 101
 - See Also* absolute position
- third-level messages, understanding 277
 - See Also* level break messages
- titles, showing and hiding 66
- totals
 - defining for columnar sections 9
 - defining for group sections 10
 - including in tabular sections 11
 - performing in-section totaling 81
 - selecting operators and operands 136
- transaction processing
 - defining for report interconnects 218
 - description of system functions 204
- typical event flow for group sections 208
- typographical conventions xxii

U

- UBE log file, locating number of optimized rows 89
- UBE, report object type 7, 20

- Underline Row Properties form 91
- underline rows
 - adding 91
 - overriding variables and variable properties 94
 - understanding 88
- Universal Batch Engine, *See* UBE
- User Defined Codes form 241
- user options
 - setting 66
 - understanding 64
- User Options form 59

V

- variable events 203
- Variable Options Selection form 175
- versions, *See* batch versions
- visual cues xxii

W

- warnings xxiii
- Welcome to the Report Design Director form 34
- Work Center
 - APIs 277
 - calling initialization API for batch error messages 283, 284
 - calling processing API for batch error messages 284
- Work Center (P012503) 275
- Work With Applications form 157, 218
- Work With Column Headings form 238
- Work With Favorites Description Translation form 222, 226, 227
- Work With Favorites form 222
- Work with Font Substitution by Language Type form 115
- Work with Fonts form 117
- Work With Forms form 157
- Work With Smart Field Templates form 268
- Work With Versions form 157, 218
- workflow, description of system functions 204
- workspace, configuring 63

