JD Edwards EnterpriseOne Tools

Report Design Aid Guide

9.2

JD Edwards EnterpriseOne Tools
Report Design Aid Guide

9.2

Part Number: E53550-06

Contents

Preface

Welcome to the JD Edwards EnterpriseOne documentation.

Documentation Accessibility

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

Access to Oracle Support

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

Related Information

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

`http://learnjde.com`

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
Italics	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`Monospace`	Monospace type indicates commands within a paragraph, URLs, code examples, text that appears on a screen, or text that you enter.
> Oracle by Example	Indicates a link to an Oracle by Example (OBE). OBEs provide hands-on, step- by-step instructions, including screen captures that guide you through a process using your own environment. Access to OBEs requires a valid Oracle account.

1 Introduction to JD Edwards EnterpriseOne Report Design Aid

JD Edwards EnterpriseOne Report Design Aid Overview

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

JD Edwards EnterpriseOne Report Design Aid Implementation

This section provides an overview of the steps that are required to implement JD Edwards EnterpriseOne Report Design Aid.

JD Edwards EnterpriseOne Report Design Aid Implementation Steps

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

This table lists the steps for the JD Edwards EnterpriseOne Report Design Aid implementation.

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

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

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

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

- Set up permissions to create OMW projects.

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

- Set up activity rules to allow you to promote projects in OMW.

  See *"Configuring Activity Rules" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

- Set up save locations to enable you to save JD Edwards EnterpriseOne objects that are not ready to be checked in.

  See *"Configuring Object Save Locations" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

- Set up default locations and printers.

  See *"Working with Report Printing Administration" in the JD Edwards EnterpriseOne Tools Report Printing Administration Technologies Guide* .

2 Understanding Report Writing

Report Writing

Oracle's JD Edwards EnterpriseOne provides fully integrated applications for managing information across an enterprise. This information includes employee data, accounts receivable and payable information, financial data, and product information. JD Edwards EnterpriseOne enables you to view and evaluate this information to make critical decisions to improve the business operation and profitability. You can also distribute this data to others with whom you do business, such as shareholders, employees, and business consultants.

JD Edwards EnterpriseOne provides reports across these systems to meet many business needs:

- Oracle's JD Edwards EnterpriseOne Financial Management
- Oracle's JD Edwards EnterpriseOne Human Capital Management
- Oracle's JD Edwards EnterpriseOne Logistics
- Oracle's JD Edwards EnterpriseOne Manufacturing

You can easily process these reports to be viewed online and in PDF, as well as export them to a spreadsheet program. However, to help you meet all of your business needs, you can create custom reports using Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA). Using this reporting tool, you can extract and present information that is vital to the business.

Reports that are used primarily to manipulate data are referred to as batch processes. Reporting and batch processing can be combined in a single report. Reports and batch processes are considered by the system as batch applications.

The Oracle's JD Edwards EnterpriseOne reporting solution includes a report design tool to create reports and batch processes, a batch engine for processing, and an output management system to output information.

Report Design Aid

You can use RDA to create a variety of simple and complex reports as well as batch processes. The interface is simple enough to use without programming expertise, yet powerful enough to create the most complex batch applications.

RDA includes a Report Director (also referred to as the Director) to guide you through the process of creating report templates. This Director presents multiple reporting options from which to choose. You can create custom Directors to aid in the creation of report templates. These Directors are configured to use report components that meet a specific reporting requirement.

After using the Director to create the initial report template, you can enhance the report by:

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

The design workspace in RDA can be configured to compliment individual work preferences. You can:

- Modify the report view options.
- Select which toolbars and windows appear in the workspace.
- Arrange windows.

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

Report Processing

You cannot process a report without a batch version. You submit the batch version for processing and can choose to process the batch version either locally or on the server. Typically, servers are faster, so processing on a server is more efficient. Once you submit a batch version, it runs without any further interaction from you. You do not interact with the report again until processing is complete.

Once you have submitted a batch version for processing, you have no control over the flow of the logic. You must make changes to the flow of logic in RDA and resubmit the batch version.

Report Components

This section discusses:

- Introduction to reports
- Report objects
- Report sections
- Report templates
- Batch versions

Introduction to Reports

A report exists as a set of specifications that are read by the Oracle's JD Edwards EnterpriseOne batch engine for processing. You can create variations of a single report using batch versions. The first step in creating a report is to create a batch application object within JD Edwards EnterpriseOne. You can accomplish this from Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW) or by accessing RDA directly from Oracle's JD Edwards EnterpriseOne Solution Explorer. You then begin designing the report using RDA. The report is actually a template from which multiple versions can be created.

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

Each report section is comprised of report objects. You can add, modify, revise, rearrange and delete report objects within a section.

Report Objects

JD Edwards EnterpriseOne is object-based. Each report template is considered a batch application with an object type of Universal Batch Engine (UBE). When you add a report object, the system creates a header record in the Object Librarian Master Table (F9860). This header record contains information about the report, such as its name and description.

Each report section is comprised of report objects. You can add different types of report objects to report sections. Not all objects are available for all section types. You can modify the properties of report objects such as:

- Font style and color
- Lines and boxes
- Text justification
- Numerical formatting

Report Sections

Report sections are the basic components of a report. Most reports include more than one section. You can use some sections for special purposes, such as performing calculations and totaling. Section types include:

- Report header and report footer sections.

  A report header section appears once at the beginning of the report. A report footer section appears once at the end of the report on its own page. You typically populate these sections using constants and variables. You can define only one of each of these sections per report.
- Page header and page footer sections.

  A page header section appears at the beginning of each page of the report. A page footer section appears at the bottom of each page of the report. You typically populate these sections using constants and variables. You can only define one of each of these sections per report.
- Detail sections

  Detail sections present the information that the report is designed to convey. The three types of detail sections are:

  - Columnar
  - Group
  - Tabular

From the Report Director in RDA, there is a fourth option for creating application reports. This option actually uses one of the three types of detail sections already mentioned—columnar, group, or tabular. The section layout of a detail section is typically populated using fields from a business view. Business views are used to access data from one or more database tables. Business views present a subset of data relevant to the immediate business requirement. Business views provide a link between the data in the database and the report that you are creating.

In addition to the business view fields that you select, you can define and add data fields to the detail report section, such as data dictionary fields, constants, and variables.

- Level break header sections

Define level break fields for use in level break header sections. Level break header sections are used to further organize data.

- Level break footer sections

  Define level break fields for use in level break footer sections. Level break footer sections are used to calculate and display totals.

A level break occurs during the processing of a report when the value of a data sequencing field, which is also defined as a level break field, changes. A set of records that contains the same value for this defined field is in the same level. For example, in an address book report that is sequenced by search type, where the search type field is also defined as a level break field, all of the records having the same search type are in the same level. (All records with a search type of E for employees are in the same level.) When the value of the search type field changes, a level break occurs. (When the search type value changes from E for employees to C for customers, a level break occurs.) Level breaks are used to group large amounts of data into more manageable units. Level break headers provide a descriptive heading prior to the associated data. Level break footers are used to include aggregates with descriptive labels in the report.

Detail Sections

Detail sections present the data required by the business need. This data is fetched from the JD Edwards EnterpriseOne database using a business view.

Within detail sections, you can:

- Attach a business view.
- Sequence data using business view fields.
- Define level breaks using data sequencing fields.
- Filter data based on designated criteria.
- Present totals.
- Attach event rules (logic statements that you create and attach to report objects).

You can include multiple detail sections in a report template. The guideline for the size of a report template is not the number of sections but rather the physical size. A report design should not exceed a physical page size of 45 inches in length and width. The RDA interface includes rulers to help you keep the report template within these parameters. Report templates that exceed this 45-inch parameter guideline might encounter problems at runtime.

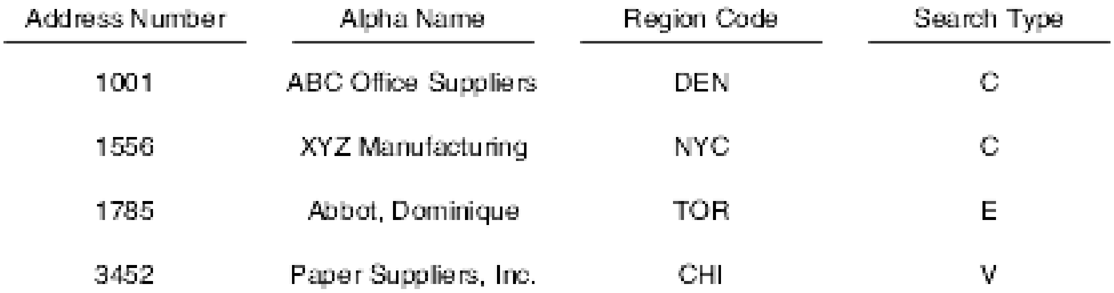
Characteristics of Columnar Sections

The columnar section format consists of column headings with rows of data under the headings. Each row is considered a record.

Each field that you add to a columnar section, includes a column heading and a column variable. The column heading describes the data. The column variable is the data that varies by record.

Due to the format of the columnar section, the column headings cannot be disconnected from their associated variables. If either the column heading or column variable is deleted, the other is also deleted.

This example shows the format of a columnar section:



Address Number	Alpha Name	Region Code	Search Type
1001	ABC Office Suppliers	DEN	C
1556	XYZ Manufacturing	NYC	C
1785	Abbot, Dominique	TOR	E
3452	Paper Suppliers, Inc.	CHI	V

You can include multiple columnar sections in a report.

You can include level break headers and level break footers in columnar section reports. The level break header appears above the column headings in columnar section reports. The level break footer displays totals and other aggregates and appears after the columnar section.

When to Use Columnar Sections

Use columnar sections when you want to display rows of data with column headings. This format is beneficial when reviewing a large number of records with specific fields of interest. For example, you want to review the salary of all the employees in the company.

You can join columnar sections to other columnar sections or to group sections. You can attach a different business view to each section. You must join the sections on common fields.

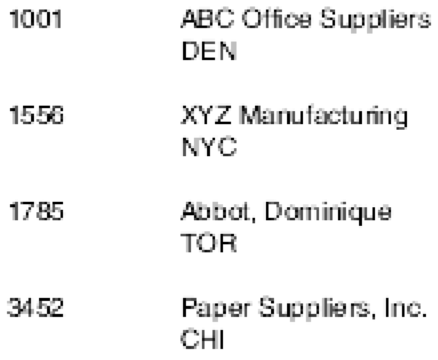
You can define columnar sections as conditional. Conditional sections are called from the report section preceding the conditional section. Conditional sections are called using event rules and process only when stated criteria is met.

See *Creating Custom Sections*.

Characteristics of Group Sections

Group sections enable you to arrange fields in a free-form layout; they are not restricted to a predefined format. The group section type is the most flexible because you can place fields anywhere in the section. Business view fields within group sections are composed of constants and variables. Initially, the constant and variable are linked; however, you can disconnect the constant from the variable to meet reporting requirements. Because of the free-form layout, group sections are ideal for creating level break footers and grand total sections.

This example shows the format of a group section:



1001	ABC Office Suppliers DEN
1556	XYZ Manufacturing NYC
1785	Abbot, Dominique TOR
3452	Paper Suppliers, Inc. CHI

You can include multiple group sections in a report.

You can include level break headers and level break footers in group section reports to print organizational headers and display totals.

When to Use Group Sections

Use group sections when a free-form layout is required to meet the reporting needs.

You can join group sections to other group sections or to columnar sections. You can attach a different business view to each section. You must join the sections on common fields.

You can define group sections as conditional. Conditional sections are called from the report section preceding the conditional section. Conditional sections are called using event rules and process only when stated criteria is met.

See *Creating Custom Sections*.

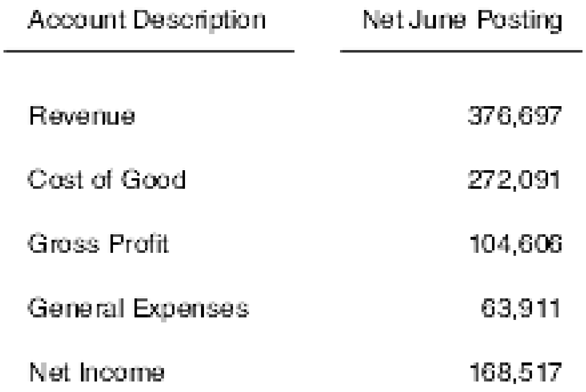
Characteristics of Tabular Sections

Tabular sections appear in the same column-and-row format as columnar sections. However, there are three major differences between columnar and tabular sections. Tabular sections offer:

- Spreadsheet functionality
- Drill-down functionality
- Row description columns

Because of the additional functionality, tabular sections are desirable for presenting numeric data that needs to be summarized with subtotals and grand totals. Typically, financial reports use tabular sections. However, tabular sections are not exclusive to financial reporting.

This example shows the format of a tabular section:



Account Description	Net June Posting
Revenue	376,697
Cost of Good	272,091
Gross Profit	104,606
General Expenses	63,911
Net Income	168,517

Within tabular sections, you can:

- Automatically calculate and display totals using level break fields.
- Automatically calculate and display grand totals.
- Define data selection at the column level.
- Define data selection at the row level.
- Define calculations at the cell level.
- Define the drill-down feature.

  The drill-down feature enables you to research values in the report by creating a link between the report output file and the associated JD Edwards EnterpriseOne application.

You can include multiple tabular sections in a report. Tabular sections processes data based on the fields you have defined as level break fields.

Tabular sections automatically include a Row Description column. This column displays descriptions for rows, based on data sequencing and level break fields. Row Description columns are typically the first columns in tabular sections.

Totaling is dynamic in tabular sections. If a column does not require totaling, you can override the totaling function. Because the totaling logic is built into tabular sections, level break footer sections are not required to provide totals. You can easily change the totaling without redesigning the report; simply change which fields you have defined as level breaks to display totals differently.

Tabular sections process differently than columnar and group sections. Therefore, there are features that are used in columnar and group section reports that are not available for tabular sections. In tabular sections you cannot:

- Add level break headers.

  The Row Description column provides this functionality.
- Add level break footers.

  Tabular sections include automatic totaling so level break footers are not needed.
- Define tabular sections as conditional.
- Join tabular sections to other sections.

Advantages of Using Tabular Sections

Advantages of using tabular sections include:

- Automatic totaling

  Totaling can be suppressed on individual columns.
- Totaling levels

  Changing which fields are defined as level break fields changes how totals are calculated.
- Audit trails

  Audit trails can be created using the drill-down feature.
- Row Description columns

  Multiple descriptions can appear in the Row Description column.

When to Use Tabular Sections

Use tabular sections when you want to display rows of data with column headings but also need to:

- Perform row level processing, such as calculations.
- Work with individual cell properties.
- Include totaling that is flexible.
- Define data selection at the column, row, or cell level.
- Calculate grand totals.
- Review detail at the application level using the drill-down feature.

Characteristics of Report Header Sections

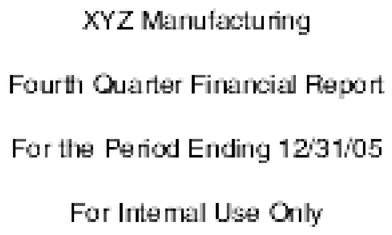
The system allows you to define only one report header in a report template, which prints once at the beginning of the report. Report headers typically include:

- Company name.
- Report title.
- Time period of the data being reported.
- Audience.

You cannot attach business views to report headers. Typically, you use data fields such as:

- Constants
- Alpha variables
- Numeric variables
- Date variables

This example shows the format of a report header:



XYZ Manufacturing

Fourth Quarter Financial Report

For the Period Ending 12/31/05

For Internal Use Only

Characteristics of Page Header Sections

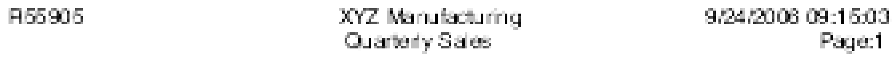
The system allows you to define only one page header in a report template, which prints once at the beginning of each page of the report. Page headers typically include:

- Company name
- Report description
- Report object name
- Date
- Time
- Page number

You have the option to allow the system to automatically generate page headers, or you can create your own. When you create your own page header you can include data fields such as:

- Constants
- Alpha variables
- Numeric variables
- Date variables
- Runtime fields

This example shows the format of a page header:



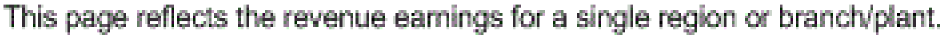
```
R55905                  XYZ Manufacturing             9/24/2008 09:15:03
                         Quarterly Sales                          Page:1
```

Characteristics of Page Footer Sections

The system allows you to define only one page footer in a report template, which prints once at the end of each page of the report. You can use page footers to present an explanation of the contents of the report. Typically, you populate page footers using data fields such as:

- Constants
- Alpha variables
- Numeric variables
- Date variables

This example shows the format of a page footer:



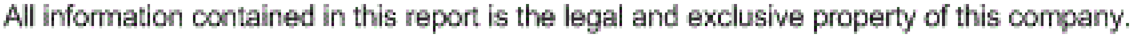
This page reflects the revenue earnings for a single region or branch/plant.

Characteristics of Report Footer Sections

The system allows you to define only one report footer in a report template, which prints once at the end of the report. You can use report footers to remind viewers that the contents of the report are for internal use only. Typically, you populate report footers using data fields such as:

- Constants
- Alpha variables
- Numeric variables
- Date variables

This example shows the format of a report footer:



All information contained in this report is the legal and exclusive property of this company.

Report Templates

Report templates are batch applications that you create in RDA. They are the master specifications of reports. These specifications describe the report to the batch engine; they define what data is used and how the data is selected, sorted, displayed, and formatted.

Because you can create multiple versions of report templates, you typically want to keep the report template generic. This means that you want to leave the data selection and data sequencing in the report template open and create batch versions with different data selection and data sequencing to meet specific business needs.

Properties defined in the report template are read by the associated batch versions. There are two exceptions to this rule:

- Batch versions that contain specifications that have been overridden.

  Specifications that are overridden in the batch version are not read from the report template.

- Report level properties that have been modified after the report template is saved and RDA is exited.

> **Note:** When you create batch versions from the Director, the system uses the current report level values. If you change the report level values prior to saving the report template and exiting RDA, the new values transfer to the version. However, if you save the report template and exit RDA, then reenter RDA and change the report level values, the modified report level values do not affect any existing versions. New batch versions that you create do reflect the modified report level values. Batch versions that you copy reflect the template specifications at the time the original version was created.

Batch Versions

Batch versions read the master specifications from the associated report template. However, batch versions typically differ slightly from the report template. For each batch version you can easily define different:

- Data selection
- Data sequencing
- 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 Batch Versions Guide* .

3 Creating Report Objects

Understanding the Report Design Process

Before you create a report, you should plan it so that it is functional, useful, and complete. A successful report design begins with planning. Determine the report requirements by:

- Asking questions of the stakeholders.
- Creating a model of the report.
- Determining which section types are required.

Surveying Stakeholders

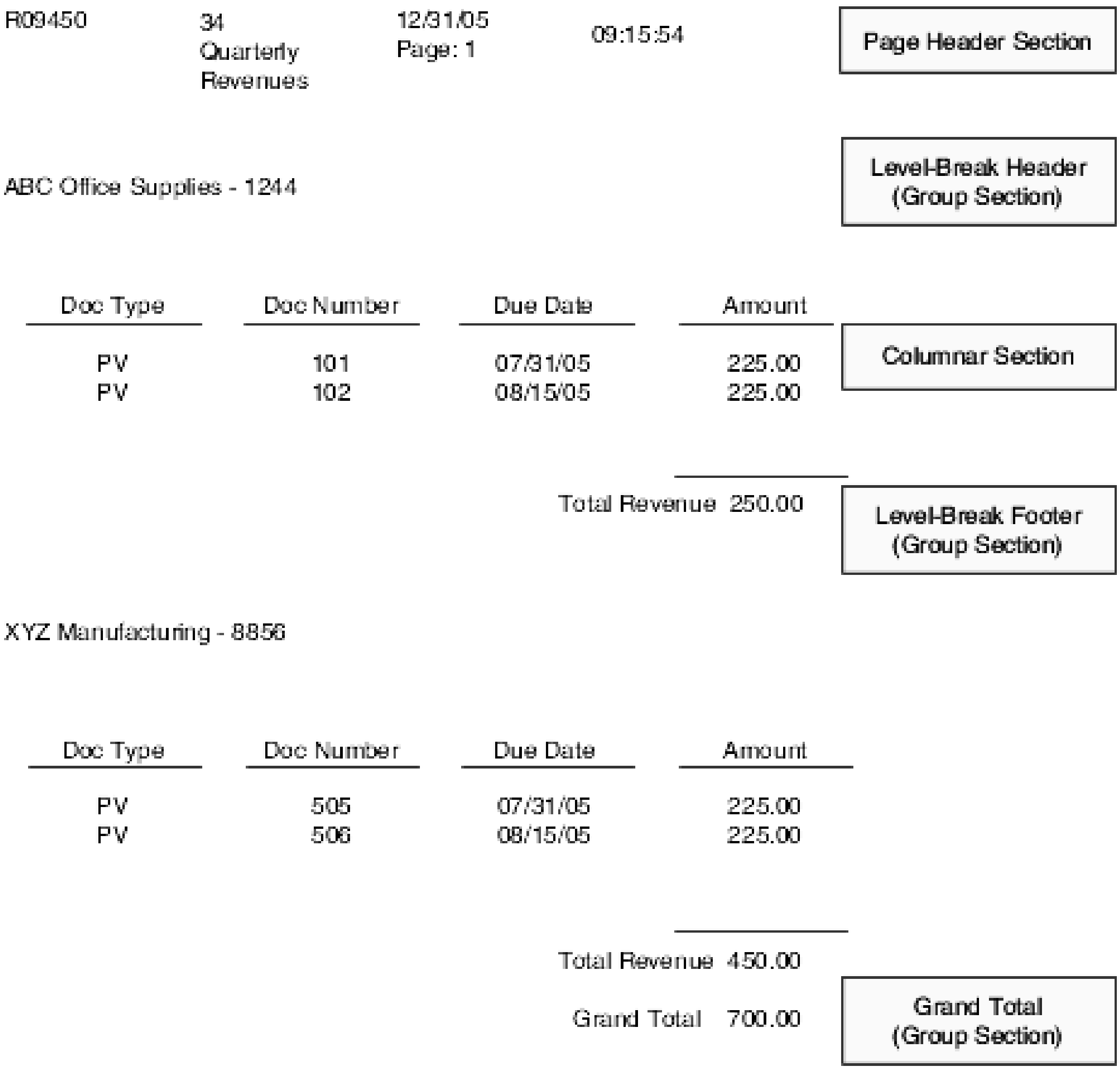
Ask these questions of the stakeholders to determine the report requirements:

- What is the purpose of the report?
- Who will use the report?
- What does the intended audience want to see in the report?
- What information in the report will come directly from the database and what information needs to be calculated?
- What are the best business views to use?
- Which data fields and records should be included in the report?
- In what order should the data be presented?
- What is the most effective format in which to present the information?
- Which reports are used to produce this information now?
- Does a report already exist that could be copied and modified?
- Will a new version of an existing report meet the business needs?
- How frequently will the report be run?
- Does the report output need to be exported to another software package using Output Stream Access (OSA) or Comma Separated Values (CSV)?

Creating Report Models

When you have determined the report requirements, create a report model. The report model helps you determine the format of the data and what section types meet the business requirements.

For example, you are creating a report to be used for invoicing or ordering. The following example illustrates a report model that meets this requirement:



```
R09450          34               12/31/05       09:15:54
                Quarterly        Page: 1
                Revenues

ABC Office Supplies - 1244

   Doc Type        Doc Number       Due Date          Amount

      PV              101           07/31/05          225.00
      PV              102           08/15/05          225.00

                                        Total Revenue  250.00

XYZ Manufacturing - 8856

   Doc Type        Doc Number       Due Date          Amount

      PV              505           07/31/05          225.00
      PV              506           08/15/05          225.00

                                        Total Revenue  450.00

                                          Grand Total  700.00
```

Page Header Section
Level-Break Header (Group Section)
Columnar Section
Level-Break Footer (Group Section)
Grand Total (Group Section)

Determining the Report Sections

After you and the report requester agree on the model, determine the sections that are required to support the model. You might conclude that the report requires multiple sections: a page header section to provide general information, a level break header section to display each company's address number and description, a columnar section to display the business data, a level break footer section to calculate and display subtotals, and a grand total section.

To determine the sections that are required for a report, consider these questions:

- Is a report header required at the beginning of the report for clear and meaningful presentation of the information?
- Is a report footer required at the end of the report to highlight an important function or fact about the report?
- Is a page header required to present information on each page of the report?

- Is a page footer required to give the reader vital information on each page of the report?
- Does an existing business view contain all of the data fields that are required by the report?
- Do you need two detail sections that use different business views so that you can include all relevant information?
- Do the two detail sections need to be joined to present the information appropriately?
- What is the best format to present the required information?
- Is the data best presented in the free-form layout of a group section?
- Is the data better presented in a columnar format using columnar or tabular sections?
- Do you need the flexibility of a tabular section to create rows in the report?
- If you use group or columnar sections, do you need to include level break footers for totaling?
- Would the automatic totaling features of a tabular section be more efficient?
- Can you use a Director template to help you create the report?
- Do you need to modify a Director template or create a new one?
- How should you sequence the business view fields and records for the report?
- How should you filter the data from the business view?
- What enhancements can you apply to report objects to make the report more useful?
  For example, do you want to emphasize the information in one column by changing the column spacing? Do you want to change the font size of the contents of a column to call attention to the information?
- Are event rules required to define conditional, mathematical, or other logic for the report?
- Is an audit trail required?

After you have finished planning the design of the report, create the report in Oracle's JD Edwards Report Design Aid (RDA).

Understanding Report Object Naming

To provide consistency for developers and users, Oracle's JD Edwards EnterpriseOne enforces standard naming conventions by which all objects are named. When you create custom objects, you should adhere to these standard naming conventions.

The JD Edwards EnterpriseOne naming conventions require that each object, whether it be a table, report, interactive application, or business view, has a unique name. The naming conventions help you identify objects and prevent users from creating objects with duplicate names. They also simplify searching for object types and for specific objects. Object names begin with one or two characters that indicate the object type. A *V* indicates that the object is a business view. An *R* indicates that the object is a batch application.

When creating multiple objects that are intended to be used together, name the objects the same, changing the first character to indicate the object type. For example, this table illustrates how to name related objects created to meet an address book business requirement:

Object Type	Object Name
Business view	V550101A

Object Type	Object Name
Processing option	T550101
Report template	R550101

See *"Understanding JD Edwards EnterpriseOne Naming Conventions", "Object Naming Conventions" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .

Creating Report Objects

This section provides overviews of default settings for reports, report objects, how to open existing reports, and how to delete report objects from multiple locations and discusses how to:

- Create report objects from RDA.
- Create report objects from Oracle's JD Edwards EnterpriseOne Object Management Workbench (OMW).
- Copy report templates from OMW.
- Open report objects from OMW.
- Delete report objects.

Understanding Default Settings for Reports

A report is defined by the system as a universal batch engine (UBE) object type. You must provide a report with:

- Object name
  Begins with an *R* to indicate that it is a report template.
- Description
  Describes the type of report and the data that it contains.
- Product code
  Indicates the system that is associated with the interactive application that uses the data, the system where the associated data resides. For example, address book data resides in system 01.

After you create a report object, you can save it and add object specifications at a later time.

When you create a new report, or other batch application, the system automatically applies default standards for some report components.

This table lists the report components that have default standards:

Report Component	Default Standard
Font	Appears in 7 point, Arial font.
Report object name	Appears in the upper-left corner of the page header.

Report Component	Default Standard
Company name	Appears in the center at the top of the page header.
Report description	Appears in the center at the top of the page header under the company name.
Date and time values	Appears in the upper-right corner of the page header.
Page number	Appears in the upper-right corner of the page header, under the date and time.

Understanding Report Objects

You can create report objects using OMW. You can also access RDA directly from Oracle's JD Edwards EnterpriseOne Solution Explorer to create report objects. The object information that you are required to enter varies depending on which method you use. When you create a new report object in RDA, you create a report template and, optionally, a batch version. The only time that you can create a batch version in RDA is when you initially create a report template. All other batch versions are created from the Oracle's JD Edwards EnterpriseOne Batch Versions (P98305) application.

When you create a new report object, you must provide the following information regarding the report:

Field	Description
Object Name	Identifies the name of the report. Begin report names with the letter *R,* followed by the product code, and then a unique identifier. The object name appears in the upper-left corner of the page header.
Description	Describes the purpose of the report. Enter a description that serves as a useful identifier, such as *General Ledger by Batch.* The description appears in the center of the page header below the company name.
Product Code	Use product codes to ensure that the custom objects remain unaltered by JD Edwards EnterpriseOne software upgrades. Product codes 55–59 are reserved for customers.
Product System Code	Identifies the system where the required data resides. Typically, the report shares the same product system code as the business view that will be used to fetch the data.
Object Use	Use this range of codes to classify the report. Object use codes 160–166 are report-related classifications.
Update Report/No Update Report	Indicates whether the report will update data in the database. Only report developers with permissions can design an update report. If the developer has permissions, the Update Report option is selected by default.

> **Note:** If you do not have permission to create a report that updates the database, the Update Report option does not appear. In this situation, the No Update Report option is selected by the system. The system administrator defines permission for the update option through JD Edwards EnterpriseOne security.

Understanding How to Open Existing Reports

You can open an existing report to make modifications. You can open and modify either a report template or a batch version. Open existing report templates either from OMW or directly from RDA. To open a report object in OMW, you must add the report object to a project. If you open and modify a report template, the changes are reflected in the associated versions, unless prohibited by overrides in the batch versions.

Open batch versions either from OMW or from Batch Versions. You can open a batch version to modify it without affecting the report template or other associated versions.

When you open an existing report, the Director is not available to assist with changes. Rather, the report template or batch version opens automatically to the RDA workspace.

> **CAUTION:** If you want to modify a report template that was shipped with JD Edwards EnterpriseOne, copy the report template first and name the copy according to the recommended naming conventions. Then modify the copied report template. This ensures that the modifications are not overwritten during a software update. The same holds true for batch versions: Do not modify batch versions that begin with XJDE or ZJDE. Copy the batch version, name the copy according to your company's requirements, and modify the copied version.

Understanding How to Delete Report Objects from Multiple Locations

You can delete batch versions and report templates from the system. If you delete a report template, all of its batch versions are deleted automatically.

You can delete report objects from these locations:

- Check-in server
- Local environment
- Save location
- Transfer locations
- All locations

  > **Note:** The objects must be checked in and you must possess the proper role and permissions to delete objects.

  Depending on the delete option that you select, and whether the system administrator has set options to allow immediate deletion, the object might not be deleted immediately. In some instances, the object is marked in bold text in the OMW project and deleted later in the development cycle.

Note:

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

Creating Report Objects from RDA

In JD Edwards Solution Explorer, select the Report Management menu (GH9111), Report Design Aid to access the JD Edwards Report Design Aid form.

1. Click New.
2. On the Create New Report form, enter a name for the report template in the Report Name field.

   Use the JD Edwards EnterpriseOne recommended naming conventions.
3. In the Description field, enter a description that identifies the purpose of the report.
4. In the Product Code field, enter a value between 55 and 59.

   Product codes 55–59 are reserved for clients for creating custom objects.
5. Select either the Update Report option or the No Update Report option, and click OK.

   If you do not have permission to create reports that update the database, the Update Report option is not visible.
6. Use the Director to design the report.
7. On the final page of the Director, select the Yes, create a version of this report option, and name the version.

Creating Report Objects from OMW

In JD Edwards Solution Explorer, from the Tools menu, select Object Management Workbench to access the Object Management Workbench form.

1. Click Find.
2. In the project view, expand the project to which the new report object will be added.
3. Click the Objects node of the project, and click Add.
4. On the Add EnterpriseOne Object to the Project form, select the Batch Application option, and click OK.
5. On the Add Object form, enter the name of the report template in the Object Name field.
6. In the Description field, enter a description that identifies the purpose of the report.
7. In the Product Code field, enter a value between 55 and 59.
8. In the Product System Code field, enter the system code that reflects where the required data resides.
9. In the Object Use field, enter a value between 160 and 166.
10. Select either the Update Report or No Update Report option, and click OK.
11. On the Batch Application Design form, select the Design Tools tab, and click Start Report Design Aid.
12. Use the Director to design the report.
13. On the final page of the Director, select the Yes, create a version of this report option, and name the version.

Note:

- *"Working with Objects", "Adding Objects to Projects" in the JD Edwards EnterpriseOne Tools Object Management Workbench Guide* .

Copying Report Templates from OMW

Access the Object Management Workbench form.

1. Click Find.
2. In the project view, expand the project where the report template that you want to copy resides.
3. Expand the Objects node, click the report, and then click Copy.
4. On the Copy Object form, enter the name of the new report template in the Copy to field.
5. In the Description field, enter a description that identifies the purpose of the report.
6. In the Product Code field, enter a value between 55 and 59.
7. In the Product System Code field, enter the system code that reflects where the required data resides
8. In the Object Use field, enter a value between 160 and 166.
9. Select either the Update Report option or the No Update Report option, and click OK.
10. On the Batch Application Design form, select the Design Tools tab, and click Start Report Design Aid.
11. Use the Director to design the report.

Opening Report Objects from OMW

Access the Object Management Workbench form.

1. Click Find.
2. In the project view, expand the project where the report object resides.

   If the report object does not reside in a project, search for the object using the OMW search tab, and move the object into a project.
3. Expand the Objects node, click the report object, and then click the Design action button.
4. If you select a report template, select the Design Tools tab on the Batch Application Design form, and click Start Report Design Aid.
5. If you select a batch version, select the Tools tab on the Batch Version Design form, and click Report Design.

   > **CAUTION:** Batch versions beginning with XJDE and ZJDE are owned by JD Edwards EnterpriseOne and should not be modified.

Deleting Report Objects

Access the Object Management Workbench form.

1. Click Find.
2. In the project view, expand the project where the report object resides.
3. Expand the Objects node, click the report, and then click Delete.
4. On the Delete of form, select one of these options and click OK:
   - Delete Object from Server.
   - Delete Object Locally.
   - Delete Object from the SAVE Location.
   - Mark Object to be Deleted from Transfer Locations.

- Remove Object from ALL Locations.

  Select this option to select all delete options.

  The deleted object appears in bold text in the project view. If the system administrator has set options to allow immediate deletion, the system deletes the object when you quit JD Edwards EnterpriseOne. Otherwise, the object is deleted when the project is advanced.

Creating Detail Sections

This section provides an overview of detail sections and discusses how to:

- Add detail sections.
- Select business views.
- Select business views from the Select Business View tab.
- Select business views from the Favorite Business Views tab.
- Add business view columns from Quick Section.
- Define section data sequencing.
- Define sort properties.
- Define section data selection.

Understanding Detail Sections

Detail sections refer to report sections that present data to meet the business requirement. Detail sections are typically populated using a business view, which links the report section to the data in the database. After you select a business view for the detail section, you can select individual fields from that business view to include in the section layout. Other types of fields can be included in the section layout and populated using event rules.

In RDA, the Director guides you through the steps of creating a report template. Initially, the Director allows you to include only one detail section in the report template. However, after you complete the initial design of the report template, you can add additional detail sections to the report.

Types of detail sections include:

- Columnar
- Group
- Tabular

> **Note:** JD Edwards EnterpriseOne supports Arabic and the right-to-left display preferences for the Microsoft Windows client, web client, and batch applications. However, when designing a report, you must design it from left to right. When you generate the report by printing it or generating a PDF, it displays information right to left, including the alignment of text and numeric strings. The system displays strings that contain a mixture of Arabic and non-Arabic characters in accordance with generally accepted bidirectional standards.

In RDA, you can create as many group, columnar, and tabular sections in a report template as required. To add one of these detail sections to an existing report template, open the report template in RDA and select the appropriate section type from the Section menu. The Director guides you through the steps of creating the new detail section.

The system automatically launches the Director when you select a section type from the menu. You can quit the Director if you prefer to design the detail section manually. Begin the detail section design by attaching a business view. Then select individual columns from that business view to include in the section layout. The options to manually design the section reside on the Section menu in the order that they are used by the Director.

Business View Selection

Business views are the link between the report and the data in the JD Edwards EnterpriseOne database. They present data fields from one or more tables and limit the column selection to columns that are relevant to the business requirement of the report. Business views also improve performance by moving only the required fields across the network.

You can select a business view for a detail section using one of two methods:

- Select Business View

  Use the Query by Example (QBE) line to filter through all business views in the system. You can filter by product code, business view name, or entire or partial description, or you can leave the QBE line empty to return all business views.

  Business view names begin with the letter V and contain a maximum of eight characters. The name is formatted as VxxzzzzA, where the variable characters represent this information:

  - xx is the product code (55–59 are reserved for clients).
  - zzzz are the characters of the primary table.
  - A indicates the sequence in which the business view was created over the primary table.

- Favorite Business Views

  Select from a list of business views that you, or other users, have defined as favorites.

No matter which method you use to select a business view, the title bar of the detail section reflects the selected business view.

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

Detail Section Layout

You select the required fields for the detail section layout from the attached business view.

When you use the Director to create detail sections, you are presented with a Section Layout form after you select the business view. This form presents all of the columns that are included in the attached business view. Select the fields that are required by the report from these available business view columns. The selected columns appear in the Selected Columns list. You can remove columns from the section layout by removing the column from the Selected Columns list.

When you design the detail section without the use of the Director, use Quick Section to select data fields from the attached business view. You should use Quick Section for the *initial* section layout only. The selected columns appear in the Selected Columns list. You can remove columns from the section layout by removing the column from the Selected Columns list.

No matter which method you use to define the section layout, when the initial layout is complete, insert additional business view columns into the detail section using the Business View Columns Browser. Remove unwanted columns by clicking the column in the section layout and selecting Delete from the Edit menu.

Data Sequencing

Use data sequencing to indicate the order in which the system should read records from the database and display them in the detail section. Data sequencing is the same as data sorting and the terms are interchangeable. For example, define section data sequencing to display records in a detail section by address book number and then by name.

JD Edwards EnterpriseOne can sort records in a report section by any column that is included in the attached business view, regardless of whether you included the column in the section layout. Other field types that you included in the section layout are not available for use in data sequencing.

Sort Properties

After you select business view fields for data sequencing, you can define sort properties for those fields. Use sort properties to indicate whether:

- Records are sorted in ascending or descending order.

  The system displays an arrow pointing up to indicate an ascending order. Click the arrow once to display an arrow pointing down to indicate a descending order.
- Sequencing fields trigger level breaks.

  Select this option to trigger a level break when the value of the field changes. A check mark indicates that the option is selected.
- Level break fields produce page breaks.

  Select this option to begin each unique value for the level break field on a new page. A check mark indicates that the option is selected.

  > **Note:** To set a field as a page break, you must first define the field as a level break.

For example, you can sort records by search type in ascending order, define search type as a level break field, and begin each unique search type on a new page.

Data Selection

Use data selection to limit the number of records that are retrieved from the JD Edwards EnterpriseOne database. Define data selection to include only specific, relevant records in the detail section. For example, define data selection to display records for hourly employees only.

JD Edwards EnterpriseOne filters data in a detail section by any column that is included in the attached business view, regardless of whether you included the column in the section layout.

Define data selection using options that are available for each of these components:

- Operator

  The operator for the first line of the data selection is *Where.* You have a choice between *And* and *Or* for additional lines of data selection.
- Left operand

  The left operand must be a column from the attached business view.

- Comparison
    - is equal to.
    - is equal to or empty.
    - is greater than.
    - is greater than or equal to.
    - is less than.
    - is less than or equal to.
    - is not equal to.
- Right operand

  The options that are available in the right operand column depend on the selection that you made in the comparison column.

  This table identifies the available options:

Option	Description
<Blank>	Enters a blank (space) value.
<Literal>	Enables you to enter specific values.
<Null>	Indicates that no value is associated with the field.
<Zero>	Enters a value of zero.
BC	Indicates a business view column.
RI	Indicates a value that is passed through a report interconnect to this report.
PC	Indicates a previous business view column.
PO	Indicates a processing option value that is available for this report.
PV	Indicates the previous value for the variable.
RC	Indicates a constant from this report.
RV	Indicates a variable from this report.

Option	Description
SV	Indicates a system variable.
SL	Indicates a system value.
TV	Indicates a text variable.
VA	Indicates an event rule variable.

When you select the Literal option for the right operand, the system presents you with a Single value form. This form includes tabs that are based on the comparison option that you selected. The tab options might include:

- Single value

  Indicates a single value such as company 00060.
- Range of values

  Indicates a range of values such as companies 00001 to 00060. When using a range of values, the only valid comparisons are *is equal to* and *is not equal to.*
- List of values

  Indicates a list of values that are not sequential. You must click Add or press Enter after each entry. To delete a value, select the value and then click Delete.

  A list of values might include several user-defined codes for search types such as C for Customers, E for Employees, and V for Suppliers. When you are using a list of values, the only valid comparisons are *is equal to* and *is not equal to.*

You can lock the data selection that you define by individual row. Click the data selection row to highlight it. Right-click the row and select Advanced from the menu. On the Data Selection Protection form, select the Locked option.

The user can clear the locked option from the JD Edwards Batch Versions application, from either the Row menu or at runtime. The locked option sends a visual to the user that the defined data selection might be required for the report to function as designed. The locked option is particularly beneficial when a section of the report adopts it's data selection from another section.

Adding Detail Sections

Access RDA.

1. Open a report template.

2. From the Section menu, select Create, and select one of these detail section types:

    - Group
    - Columnar
    - Tabular

3. Use the Director to design the section.

Selecting Business Views

Open a report template in RDA.

1. Click inside the detail section of the report.
2. From the Section menu, click Select Business View.
3. On the Business View Director, select a tab:

    - Select Business View
    - Favorite Business Views

4. Proceed to the appropriate task.

Selecting Business Views from the Select Business View Tab

From the Section menu in RDA, click Select Business Views to access the Business View Director.

1. Select the Select Business View tab.
2. On the Select Business View form, enter the business view name on the Object Name QBE line, and click Find or press Enter.
3. Select a business view, and click OK.

Selecting Business Views from the Favorite Business Views Tab

From the Section menu in RDA, click Select Business Views to access the Business View Director.

1. Select the Favorite Business Views tab.
2. From the Favorites view, expand each folder until you locate the required business view.
3. Select the Description tab to view a brief description of the business view that you have selected.
4. Select the Columns tab to view the data fields that are included in the business view that you have selected.
5. When you are certain that you have selected the appropriate business view, click OK.

> **Note:**
>
> - *Setting Up Business Views as Favorites*

Adding Business View Columns Using Quick Section

Open a report template in RDA.

1. Click a detail section of the report that includes an attached business view but no fields.
2. From the Section menu, select Quick Section.
3. On the Quick Section form, click a required column in the Available Business View Columns list, and perform one of these tasks:
    - Click the right arrow to move the column to Selected Columns.
    - Drag the column to Selected Columns.

      You can move multiple columns to Selected Columns by using the Shift or Ctrl keys.
    - Click the right-double arrow to move all columns to Selected Columns.
4. To remove columns, click a column in the Selected Columns list and perform one of these tasks:
    - Click the left arrow or press Delete to remove the selected column.

      You can remove multiple columns by using the Shift or Ctrl keys.
    - Click the left-double arrow to remove all columns.
5. To change the order in which the columns appear on the report, click a column from the Selected Columns list and perform one of these tasks:
    - Click the up arrow or down arrow to move the selected column up or down one position in the list.
    - Drag the selected column to another position in the list.
    - Click the up-double arrow or the down-double arrow to move the selected column to the top or bottom of the list.
6. Complete the layout of the section, and click OK.

Defining Section Data Sequencing

Open a report in RDA.

1. Click a detail section of the report in which the section layout is defined.
2. From the Section menu, select Define Data Sequence.
3. On the Data Sequencing form, select the Section Data Sequencing tab.
4. To select columns for data sequencing, click a column in the Available Columns list and perform one of these tasks:
    - Click the right arrow to move the column to Selected Columns.
    - Drag the column to Selected Columns.

      You can move multiple columns to Selected Columns by using the Shift or Ctrl key.
    - Click the right-double arrow to move all columns to Selected Columns.

5. To remove columns from the data sequencing, click a column in the Selected Columns list and perform one of these tasks:
   - Click the left arrow or press Delete to remove the selected column.

     You can remove multiple columns by using the Shift or Ctrl key.
   - Click the left-double arrow to remove all columns.
6. To change the order of the fields that are used to sort the section data, click a column in the Selected Columns list and perform one of these tasks:
   - Click the up arrow or down arrow to move the selected column up or down one position in the list.
   - Drag the selected column to another position in the list.
   - Click the up-double arrow or the down-double arrow to move the selected column to the top or bottom of the list.

Defining Sort Properties

From the Section menu in RDA, select Define Data Sequence to access the Data Sequencing form.

1. Select the Define Sort Properties tab.
2. Complete the Sort Order field for each sort column.
3. Click these fields as appropriate, and click OK:
   - Level Break
   - Page Break

     You must define a field as a level break before you can define it as a page break.

Defining Section Data Selection

Open a report in RDA.

1. Click a detail section of the report that has a business view attached.
2. From the Section menu, select Define Data Selection.
3. On the Data Selection form, click in the Operator column of the detail area and select *Where* for the first set of criteria.
4. Click in the Left Operand column to display the list of available business view fields, and perform one of these tasks:
   - Scroll through the list until you find the required business view field, and then double-click the field to populate the Left Operand column.
   - Begin entering the business view field name in the Left Operand column.

     The system presents the first business view field in the list that matches the criteria. Continue entering the field name until the required field appears, and then double-click the field.

     When you double-click the business view field to select it for the Left Operand column, the list in the Comparison column appears.
5. Select an operator in the Comparison column.

When you double-click the comparison operator, the list in the Right Operand column appears.

6. In the Right Operand column, select from the available objects, special values, and variables.
7. To delete a line of criteria on the Data Selection form, click the row header to highlight the row, and then click the Delete button at the top of the form.
8. To change the order of the criteria, click the row header to highlight the row, and then click the up arrow or down arrow button.

4 Using the Report Director

Understanding the Report Director

The Director provides a quick start to designing reports by guiding you through a linear process for setting up the basic elements. As you proceed through the Director, you can select elements to include in the report template, such as:

- Header and footer sections.
- Type of detail section that best meets the business requirement.

  Using the Director, you can include only one detail section in the report template. You can add more detail sections to the report template after you complete the initial design and quit the Director. For example, you can add:
- The appropriate business view to fetch the required data.
- Business view fields that you want to include in the detail section layout.

Using the Director, you can define which records to include in the detail section and how to sort those records. Based on the elements that you select, the Director creates a report template and opens it in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA). You can then use RDA to format the data, enhance the report, and add additional detail sections.

When you create an application report, the Director uses templates to guide you through the design. Director templates define default criteria to meet a specific business requirement, such as:

- Detail section type
- Business view
- Processing options
- Data selection
- Data sequencing
- Smart fields
- Drill down

When you select an application report template from the Director, the Director reads the template specifications, which are stored in Oracle's JD Edwards EnterpriseOne tables, and presents the default criteria. Several Director templates are shipped with the JD Edwards EnterpriseOne software. You can create custom templates from the Oracle's JD Edwards EnterpriseOne Report Director Templates (P91400) application.

On the welcome form of the Report Design Director, you can select from these section options:

- Report header

  Creates an empty section that must be completed manually.
- Page header

  Creates a section that can be populated automatically by the system or entered manually.
- Columnar

  Creates a detail section that provides a columnar format that includes column headings with rows of data.

- Group

  Creates a detail section that provides a free-form layout that enables you to place fields exactly where you want them.
- Tabular

  Creates a detail section that provides a columnar format with additional features that are unique to the tabular section type.
- Application Reports

  Creates a detail section that uses Report Director templates. The system displays Financial Reports as the default template. Use the drop-down list box to select from available Report Director templates.
- Page footer

  Creates an empty section that must be completed manually.
- Report footer

  Creates an empty section that must be completed manually.

If you want to design a report without the assistance of the Director, you can click Finish or Cancel to quit the Director. When you click Finish, the system accepts all of the selections that you have already made using the Director. When you click Cancel, the system ignores any selections that you have made. You can then build the report manually using options on the Section menu in RDA.

Creating Columnar Sections

This section provides an overview of columnar sections, lists the prerequisite, and discusses how to:

- Include sections in columnar section reports.
- Select business views in columnar section reports.
- Select business view columns in columnar section reports.
- Define section data sequencing in columnar section reports.
- Define sort properties in columnar section reports.
- Include records in columnar section reports.
- Create batch versions of columnar section reports from the Director.
- Create an example columnar section report.

Understanding Columnar Sections

The columnar section report presents information from a business view in a columnar format. Each data field from the business view is a column and each record is a row.

Each business view field in a columnar section consists of a:

- Column heading

  Description of the field, such as Address Number.

- Column variable

  Data that might change for each new record, such as the address number value, for example, 1001, 1202.

You can use the Director to guide you through the process of creating columnar section reports. The Director prompts you to select basic elements regarding the structure and content of the report. The companion Navigation Assistant tracks where you are in the report development process. You can right-click the Navigation Assistant to hide it for the current design process.

After you finish creating the initial report template using the Director, you can enhance it by using additional features of RDA.

> **Note:**
> - *Setting Up Business Views as Favorites*.
> - *Enhancing Reports Using Advanced Functionality*.

Prerequisite

Before you can select sections to include in a columnar section report, you must create a batch application object. The system automatically opens the welcome form of the Director as the last step in creating a batch application object.

See *Creating Report Objects*.

Including Sections in Columnar Section Reports

Access the Welcome to the Report Design Director form.

1. Select from the header and footer options that are available.

   Typically, you include a page header section as a standard element of a report. All other header and footer sections are optional.
2. Select the Columnar option, and click Next.

   > **Note:** If you do not select to include a page header in the report, the Business View Selection Option form appears, and you can bypass the rest of the steps in this task.

3. On the Page Header Details form, select these options to allow the system to automatically populate the page header section, and then click Next:
   - Automatically add the default informational fields shown below to my page header section.
   - Automatically add the default informational fields Page n or Total to my page header section.

     Select this option to include the Page n of Total field in the page header. You must also select the first option if you want to include all fields in the page header.

     > **Note:** After you complete the initial design of the report using the Director, you can add or delete fields from the page header. If you do not select either of these page header options, the system creates an empty page header. You can manually add fields to the page header from the Section menu in RDA.

Selecting Business Views in Columnar Section Reports

Click Next on the Welcome to the Report Design Director form to access the Business View Selection Option form.

1. Select one of these options to locate an appropriate business view, and click Next:
   - I'd like help in finding an appropriate business view.
   - I'll find a business view myself.
2. If you selected the option I'd like help in finding an appropriate business view, expand the favorites folders on the Favorites Business Views form until you locate an appropriate business view, select the business view, and click Next.
3. If you selected the option I'll find a business view myself, locate an appropriate business view using the QBE (query by example) line on the Select Business View form, and click Next.

Selecting Business View Columns in Columnar Section Reports

Click Next on the Select Business View form or the Favorite Business Views form to access the Section Layout form.

1. Select the columns to include in the columnar section from the Available Business View Columns list, and move the columns to Selected Columns.

   Columns appear in the columnar section from left to right in the order that they are listed in Selected Columns.

   You can select multiple columns using the Ctrl or Shift keys. You can also drag each column individually to Selected Columns, or you can click the right-double arrow to move all of the columns from the Available Business View Columns list to Selected Columns.
2. To remove columns from the section layout, select a column in the Selected Columns list and move the column to the Available Business View Columns list.

   You can also click the left-double arrow to remove all of the columns from the Selected Columns list.
3. To change the order in which the columns appear in the columnar section, select a column in the Selected Columns list and click the up arrow or down arrow to move the selected column up or down in the list one line at a time.

   You can also drag a column to a new location in the list, or you can click the up-double arrow or down-double arrow to move the selected column to the top or bottom of the list.
4. When the section layout is complete, click Next.

Defining Section Data Sequencing in Columnar Section Reports

Click Next on the Section Layout form to access the Section Data Sequencing form.

1. Select columns for sequencing from the Available Columns list, and move them to Selected Columns.

   You can select multiple columns, remove columns, and change the order of columns in the same manner as on the Section Layout form.

   > **Note:** To define sort properties in the next task, you must select columns in this task.

2. When the data sequencing for the section is complete, click Next.

Defining Sort Properties in Columnar Section Reports

Click Next on the Section Data Sequencing form to access the Define Sort Properties form.

1. Click the arrow in the Sort Order field until it reflects your preference of ascending or descending order.
2. To specify level breaks and page breaks for specific columns, click these fields next to the appropriate column, and then click Next:

    - Level Break
    - Page Break

    You must define a column as a level break before you can define it as a page break.

Including Records in Columnar Section Reports

Click Next on the Define Sort Properties form to access the Data Selection form.

1. Click the Operator field in the detail area and select **Where** as the operator.
2. In the Left Operand field, select from the list of available business view columns.
3. In the Comparison field, select from the list of available options.
4. In the Right Operand field, select from the list of available objects.
5. To delete a line of criteria, click the row header to highlight the row, and then click the Delete button at the top of the form.
6. To change the order of the criteria, click the row header to highlight the row and then click the up arrow or the down arrow button.
7. When the selection criteria is complete, click Next.

Creating Batch Versions of Columnar Section Reports from the Director

Click Next on the Data Selection form to access the Finish form of the Director.

1. To allow the system to automatically generate a batch version of the report template, select the Yes, create a version of this report option.
2. Enter the name of the version in the field beneath the Yes, create a version of this report option.
3. To review the selections, click Back to move backwards through the Director forms.

    On the Navigation Assistant, you can also click the name of the form that you want to review.
4. When you are satisfied, click Finish.

    > **Note:** When you click Finish, you can no longer access the Director for this report. Before you click Finish, you can review the selections on all forms of the Director.

Creating an Example Columnar Section Report

Use the Director to create this example columnar section report which is used for annual salary reviews. A columnar section format was selected for this report because the information is best presented in rows and columns. The columnar format makes reviewing the salary column by employee easier. The V060116A - Employee Master business view is attached to the columnar section, and these business view columns were selected for the section layout:

- Address Number
- Name - Alpha
- Business Unit - Home
- Pay Class (H/S/P)
- Date - Original Employment
- Rate - Salary, Annual

Access RDA.

1. From the File menu. select New.
2. On the Create New Report form, enter **R55060116** in the Report Name field.
3. Enter **Annual Salary Review** in the Description field.
4. Enter **55** in the Product Code field.
5. Select the No Update Report option, and click OK.
6. On the Welcome to the Report Design Director form, select the Page Header and Columnar sections, and click Next.
7. On the Page Header Details form, click Next to accept the default.
8. On the Business View Selection Option form, select the option I'll find a business view myself, and click Next.
9. On the Select Business View form, locate and select the **V060116A - Employee Master** business view, and click Next.
10. On the Section Layout form, select these columns in the order listed, and move them to Selected Columns:
    - Address Number
    - Name - Alpha
    - Business Unit - Home
    - Pay Class (H/S/P)
    - Date - Original Employment
    - Rate - Salary, Annual
11. When the selected columns are arranged as indicated, click Next.
12. On the Data Sequencing form, move the Name - Alpha column to Selected Columns, and click Next.
13. On the Define Sort Properties form, sort the data in ascending order, and click Next.
14. On the Data Selection form, click Next.

    No data selection is required for this example.
15. On the Finish form, select Yes, create a version of this report, enter **VER0001** as the version name, and then click Finish.
16. On the JD Edwards Report Design Aid form, click Save on the toolbar to save the report.
17. Select the Preview tab to preview the report.
18. From the File menu, select Exit to quit RDA.

Creating Group Sections

This section provides an overview of group sections, lists the prerequisite, and discusses how to:

- Include sections in group section reports.
- Select business views in group section reports.
- Select business view columns in group section reports.
- Define section data sequencing in group section reports.
- Define sort properties in group section reports.
- Include records in group section reports.
- Create batch versions of group section reports from the Director.
- Create an example group section report.

Understanding Group Sections

Group sections provide flexibility in arranging the required data. A group of fields in a group section represents data from one record. You must position the fields so that you can easily to determine where each new record begins.

Each business view field in a group section consists of a:

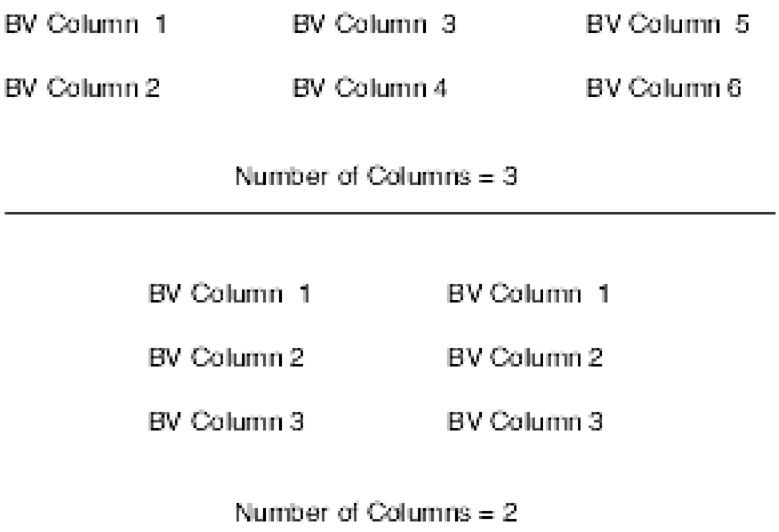
- Constant

  Description of the field, such as Address Number.
- Variable

  Data that might change for each new record, such as the address number value, for example, 1001, 1202.

When you select business view columns for group section layouts, you must specify how many of the fields to list across the page before beginning the next vertical grouping. The Number of Columns field appears below the Available Business View Columns list. The Number of Columns value determines how RDA organizes fields in a group section. For example, if you select six fields and enter **3** in the Number of Columns field, the fields in the section are organized in three columns, each including two business view fields. The first two fields appear in the first column in the same order that they appear in the Selected Columns list. The next two fields appear in the second column in the same order that they appear in the Selected Columns list. The last two fields appear in the third column in the same order that they appear in the Selected Columns list. If you enter **2** in the Number of Columns field, the business view fields are arranged in two columns. Each column includes three business view fields. The Number of Columns default value is **2.**

This example illustrates how the columns are organized in a group section based on the value that is entered in the Number of Columns field:



Note: The term *business view column* might be confusing with regard to group sections because group sections are not organized in a columnar format. A business view column refers to the column name from the table in the database.

After you create a new report object, use the Director to design a group section report. The Director prompts you to select basic elements regarding the structure and content of the report. The companion Navigation Assistant tracks where you are in the report development process. You can right-click the Navigation Assistant to hide it for the current design process.

After you have finished creating the initial report template using the Director, you can enhance it by using additional features of RDA.

Note:

- *Enhancing Reports Using Basic Functionality*.
- *Enhancing Reports Using Advanced Functionality*.

Prerequisite

Before you can select sections to include in a group section report, you must create a batch application object. The system automatically opens the welcome form of the Director as the last step in creating a batch application object.

See *Creating Report Objects*.

Including Sections in Group Section Reports

Access the Welcome to the Report Design Director form.

1. Select from the header and footer options that are available.

Typically, you include a page header section as a standard element of a report. All other header and footer sections are optional.

2. Select the Group option, and click Next.

> **Note:** If you do not select to include a page header in the report, the Business View Selection Option form appears, and you can bypass the rest of the steps in this task.

3. On the Page Header Details form, select these options to automatically populate the page header section, and then click Next:
   - Automatically add the default informational fields shown below to my page header section.
   - Automatically add the default informational fields "Page n or Total" shown below to my page header section.

     Select this option to include the Page n of Total field in the page header. You must also select the first option if you want to include all fields in the page header.

Selecting Business Views in Group Section Reports

Click Next on the Welcome to the Report Design Director form to access the Business View Selection Option form.

1. Select one of these options to locate an appropriate business view, and click Next:
   - I'd like help in finding an appropriate business view.
   - I'll find a business view myself.
2. If you selected the option I'd like help in finding an appropriate business view, expand the favorites folders on the Favorites Business Views form until you locate an appropriate business view, select the business view, and click Next.
3. If you selected the option I'll find a business view myself, locate an appropriate business view using the QBE line on the Select Business View form, and click Next.

Selecting Business View Columns in Group Section Reports

Click Next on the Select Business View or Favorite Business Views form to access the Section Layout form.

1. Select the columns to include in the group section from the Available Business View Columns list, and move the columns to Selected Columns.
2. To remove columns from the section layout, select a column in the Selected Columns list, and move it to the Available Business View Columns list.
3. To change the order in which the columns appear in the columnar section, select a column in the Selected Columns list, and move the selected column up or down in the list one line at a time.
4. Enter a value in the Number of Columns field, and click Next.

Defining Section Data Sequencing in Group Section Reports

Click Next on the Section Layout form to access the Data Sequencing form.

1. Select columns for sequencing from the Available Columns list, and move them to Selected Columns.

You can select multiple columns, remove columns, and change the order of columns in the same manner as on the Section Layout form.

> **Note:** To define sort properties in the next task, you must select columns in this task.

2. When the data sequencing for the section is complete, click Next.

Defining Sort Properties in Group Section Reports

Click Next on the Section Data Sequencing form to access the Define Sort Properties form.

1. Click the arrow in the Sort Order field until it reflects your preference of ascending or descending order.
2. To specify level breaks and page breaks for specific columns, click these fields next to the appropriate column, and then click Next:
   - Level Break
   - Page Break

   You must define a column as a level break before you can define it as a page break.

Including Records in Group Section Reports

Click Next on the Define Sort Properties form to access the Data Selection form.

1. Click the Operator field in the detail area, and select **Where** as the operator.
2. In the Left Operand field, select from the list of available business view columns.
3. In the Comparison field, select from the list of available options.
4. In the Right Operand field, select from the list of available objects.
5. To delete a line of criteria, click the row header to highlight the row, and then click the Delete button at the top of the form.
6. To change the order of the criteria, click the row header to highlight the row, and then click the up arrow or down arrow button.
7. When the selection criteria is complete, click Next.

Creating Batch Versions of Group Section Reports from the Director

Click Next on the Data Selection form to access the Finish form of the Director.

1. To allow the system to automatically generate a batch version of the report template, select the Yes, create a version of this report option.
2. Enter the name of the version in the field beneath the Yes, create a version of this report option.
3. To review the selections, click Back to move backwards through the Director forms.

   On the Navigation Assistant, you can click the name of the form that you want to review.
4. When you are satisfied, click Finish.

> **Note:** When you click Finish, you can no longer access the Director for this report. Before you click Finish, you can review the selections on all forms of the Director.

Creating an Example Group Section Report

Use the Director to create this example group section report, which is used for reviewing inventory on hand. A group section format was selected for this report because the information for a specific inventory item is easily reviewed in a group. The data selection filters the data so that inventory items are included in the report for only one business unit, business unit 27. The V41021E - Item Location, Item Master Join business view is attached to the group section, and these business view columns were selected for the section layout:

- Location.
- Item Number - Short.
- Primary Location (P/S).
- Category G/L.
- Quantity on Hand - Primary Units.
- Quantity on Backorder.
- Business Unit.

Access RDA.

1. From the File menu, select New.
2. On the Create New Report form, enter **R5541021** in the Report Name field.
3. Enter **Inventory on Hand** in the Description field.
4. Enter **55** in the Product Code field.
5. Select the No Update Report option, and click OK.
6. On the Welcome to the Report Design Director form, select the Page Header and Group sections, and click Next.
7. On the Page Header Details form, click Next to accept the default.
8. On the Business View Selection Option form, select the option I'll Find a business view myself, and clickNext.
9. On the Select Business View form, locate and select the **V41021E - Item Location, Item Master Join** business view, and click Next.
10. On the Section Layout form, select these columns in the order listed, and move them to Selected Columns:
    - Location.
    - Item Number - Short.
    - Primary Location (P/S).
    - Category G/L.
    - Quantity on Hand - Primary units.
    - Quantity on Backorder.
    - Business Unit.
11. When the selected columns are arranged as indicated, enter **3** in the Number of Columns field, and click Next.
12. On the Data Sequencing form, move the Location and Item Short columns to Selected Columns, and click Next.
13. On the Define Sort Properties form, sort the data for both fields in ascending order, and click Next.
14. On the Data Selection form, complete these fields:

Field	Value
Operator	**Where**
Left Operand	**Business Unit (F41021) (MCU) (BC)**
Comparison	**is equal to**
Right Operand	**<Literal>**

15. Select the Single value tab on the Single Value form, enter **27** in the Business Unit field, and click OK.
16. On the Data Selection form, click Next.
17. On the Finish form, select Yes, create a version of this report, enter **VER0001** as the version name, and click Finish.
18. On the JD Edwards Report Design Aid form, click Save to save the report.
19. Select the Preview tab to preview the report.
20. From the File menu, select Exit to quit RDA.

Creating Tabular Sections

This section provides an overview of tabular sections, lists the prerequisite, and discusses how to:

- Include sections in tabular section reports.
- Select business views in tabular section reports.
- Select business view columns in tabular section reports.
- Define section data sequencing in tabular section reports.
- Define sort properties in tabular section reports.
- Include records in tabular section reports.
- Create batch versions of tabular section reports from the Director.
- Create an example tabular section report.

Understanding Tabular Sections

The tabular section report presents information from a business view in a columnar format, similar to columnar sections; however, tabular sections provide additional features such as:

- Drill down
  Enables you to link the data in tabular sections to the data in the associated application. This enables users to click the data in the online PDF and review the entries in the application. This feature is especially beneficial to auditors and other users who need to view how an amount was derived.
  See *Working with the Drill Down Feature*.

- Row Description columns

  Displays multiple fields, and includes *all* fields that you define as level break fields. The description of the level break fields appears when a description business function is attached to the data item in data dictionary; otherwise, the field value appears.

  See *Working with Row Description Columns*.
- Automatic totaling

  Calculated by the system based on the fields that you define as level break fields.
- Spreadsheet functionality

  Enables you to add data to tabular sections using rows, defining each row individually. You can override individual cells of the rows by selecting Cell Mode from the Cell menu.

  See *Working with Rows*.

  See *Overriding Cells and Cell Properties*.

You can use the Director to guide you through the process of creating tabular section reports. The Director prompts you to select basic elements regarding the structure and content of the report. The companion Navigation Assistant tracks where you are in the report development process. You can right-click the Navigation Assistant to hide it for the current design process.

After you finish creating the initial report template using the Director, you can enhance it by using additional features of RDA, including features that are unique to tabular sections.

Note:

- *Working with Objects Unique to Tabular Sections*
- *Working with the Drill Down Feature*.
- *Working with Smart Fields*.

Prerequisite

Before you can select sections to include in a tabular section report, you must create a batch application object. The system automatically opens the welcome form of the Director as the last step in creating a batch application object.

See *Creating Report Objects*.

Including Sections in Tabular Section Reports

Access the Welcome to the Report Design Director form.

1. Select from the header and footer options that are available.

Typically, you include a page header section as a standard element of a report. All other header and footer sections are optional.

2. Select the Tabular option, and click Next.

   > **Note:** If you do not include a page header in the report, the Business View Selection Option form appears, and you can bypass the rest of the steps in this task.

3. On the Page Header Details form, select these options to allow the system to automatically populate the page header section, and then click Next:

   - Automatically add the default informational fields shown below to my page header section.
   - Automatically add the default informational field "Page n of Total" shown below to my page header section.

     Select this option to include the Page n of Total field in the page header. You must also select the first option if you want to include all fields in the page header.

Selecting Business Views in Tabular Section Reports

Click Next on the Welcome to the Report Design Director form to access the Business View Selection Option form.

1. Select one of these business view options, and click Next:

   - I'd like help in finding an appropriate business view.
   - I'll find a business view myself.
2. If you selected the option I'd like help in finding an appropriate business view, expand the favorites folders on the Favorites Business Views form until you locate an appropriate business view, select the business view, and click Next.
3. If you selected the option I'll find a business view myself, locate an appropriate business view using the QBE line on the Select Business View form, and click Next.

Selecting Business View Columns in Tabular Section Reports

Click Next on the Select Business View for or the Favorite Business Views form to access the Section Layout form.

1. From the Available Business View Columns list, select the columns to include in the columnar section, and move them to Selected Columns.

   Columns appear in the tabular section from left to right in the order that they are listed in Selected Columns.
2. To remove columns from the section layout, select a column in the Selected Columns list, and move it to the Available Business View Columns list.
3. To change the order in which the columns appear in the columnar section, select a column in the Selected Columns list, and move the selected column up or down in the list one line at a time.
4. When the section layout is complete, click Next.

Defining Section Data Sequencing in Tabular Section Reports

Click Next on the Section Layout form to access the Data Sequencing form.

1. Select columns for sequencing from the Available Columns list, and move them to Selected Columns.

   You can select multiple columns, remove columns, and change the order of columns in the same manner as on the Section Layout form.

   > **Note:** To define sort properties in the next task, you must select columns in this task.

2. When the data sequencing for the section is complete, click Next.

Defining Sort Properties in Tabular Section Reports

Click Next on the Section Data Sequencing form to access the Define Sort Properties form.

1. Click the arrow in the Sort Order field until it reflects your preference of ascending or descending order.
2. To specify level breaks and page breaks for specific columns, click these fields next to the appropriate column, and then click Next:

   - Level Break
   - Page Break

     You must define a column as a level break before you can define it as a page break.

Including Records in Tabular Section Reports

Click Next on the Define Sort Properties form to access the Data Selection form.

1. Click the Operator field in the detail area, and select **Where** as the operator.
2. In the Left Operand field, select from the list of available business view columns.
3. In the Comparison field, select from the list of available options.
4. In the Right Operand field, select from the list of available objects.
5. To delete a line of criteria, click the row header to highlight the row, and then click the Delete button at the top of the form.
6. To change the order of the criteria, click the row header to highlight the row, and then click the up arrow or down arrow button.
7. When the selection criteria is complete, click Next.

Creating Batch Versions of Tabular Section Reports from the Director

Click Next on the Data Select form to access the Finish form of the Director.

1. To allow the system to automatically generate a batch version of the report template, select the Yes, create a version of this report option.
2. Enter the name of the version in the field beneath the Yes, create a version of this report option.
3. To review the selections, click Back to move backwards through the Director forms.

   On the Navigation Assistant, you can click the name of the form that you want to review.
4. When you are satisfied, click Finish.

   > **Note:** When you click Finish, you can no longer access the Director for this report. Before you click Finish, you can review the selections on all forms of the Director.

Creating an Example Tabular Section Report

Use the Director to create this example tabular section report, which is used to review outstanding purchase orders by business unit. A tabular section format was selected for this report for the automatic totaling and Row Description column features that are available in tabular sections. The V4311A - Purchase Order Detail Browse business view is attached to the tabular section, and these columns were selected for the section layout:

- Description
- Business Unit
- Order Type
- Amount Open

The report is organized by company and displays item descriptions. The data is filtered to include only purchase orders (as opposed to items that were ordered by other methods, such as purchase requisitions) for stocked parts. These parts carry a balance and are not yet closed

Access RDA.

1. From the File menu, select New.
2. On the Create New Report form, enter **R554311** in the Report Name field.
3. Enter **Outstanding Purchase Orders** in the Description field.
4. Enter **55** in the Product Code field.
5. Select the No Update Report option and click OK.
6. On the Welcome to the Report Design Director form, select the Page Header and Tabular sections, and click Next.
7. On the Page Header Details form, click Next to accept the default.
8. On the Business View Selection Option form, select the option I'll find a business view myself, and then click Next.
9. On the Select Business View form, locate and select the **V4311A - Purchase Order Detail Browse** business view, and click Next.
10. On the Section Layout form, select these columns in the order listed and move them to Select Columns:

- Description

  The system has already added a Description column for you in Selected Columns. This is the Row Description column.
- Business Unit
- Order Type
- Amount - Open

11. When the selected columns are arranged as indicated, click Next.
12. On the Data Sequencing form, move the Order Company column (from table F4311) and the 2nd Item Number column to Selected Columns, and click Next.

    > **Note:** The data in this tabular section is sorted on business view columns that are not included in the section layout.

13. On the Define Sort Properties form, sort the data for both columns in ascending order.
14. Select the Level Break option for both fields, and click Next.

    The Row Description column includes both of these columns, and the system calculates totals for open orders each time that the system displays a new record.
15. To filter data to include only open purchase orders of stock items that have an amount open, complete these criteria on the Data Selection form, and then click Next:

    This filter statement meets the criteria that the open purchase orders include an open amount:

Column	Value
Operator	**Where**
Left Operand	**Amount - Open (F4311) (AOPN) (BC)**
Comparison	**is greater than**
Right Operand	*<Zero>* Do not enter a value of zero as a literal value.

> **Note:** Each succeeding line of the filter is connected with an **And** operator. You must use **And** because every data item must meet *all* of the criteria to be included in the section.

This filter statement meets the criteria that the records are purchase orders:

Column	Value
Operator	**And**

Column	Value
Left Operand	**Order Type (F4311) (DCTO) (BC)**
Comparison	**is equal to**
Right Operand	**OP**

This filter statement meets the criteria that the purchase orders are still open, or not complete:

Column	Value
Operator	**And**
Left Operand	**Status Code - Next (F4311) (NXTR) (BC)**
Comparison	**is not equal to**
Right Operand	**999** 999 is a literal value for Complete.

This filter statement meets the criteria that the purchase orders include stock items only:

Column	Value
Operator	**And**
Left Operand	**Line Type (F4311) (LNTY) (BC)**
Comparison	**is equal to**
Right Operand	- **S** S is a literal value for Stock Item.

16. On the Finish form, select the No, I will create a version of this report later option, and click Finish.
17. On the JD Edwards Report Design Aid form, click Save to save the report.
18. Select the Preview tab to preview the report.

19. From the File menu, select Exit to quit RDA.

Creating Application Reports

This section provides overviews of application reports, smart fields, and calculation columns, lists the prerequisite, and discusses how to:

- Select application report templates.
- Select business views in application reports.
- Select smart fields in application reports.
- Create calculation columns in application reports.
- Define section data sequencing in application reports.
- Define section data sequencing using the Advanced option.
- Include records in application reports.
- Define additional properties in application reports.
- Create batch versions of application reports from the Director.

Understanding Application Reports

You must create application reports using the Director. The Director uses Report Director templates that define default criteria for creating report templates over specific JD Edwards EnterpriseOne systems. When you select one of the templates from the Director, the system reads the template specifications and presents the default criteria through the Director forms. You can modify these templates or create custom templates using the Oracle's JD Edwards EnterpriseOne Director Templates application (P91400). Application reports can include any report section type, as defined by the Director template.

Director templates enable you to use smart fields, which are data dictionary items (glossary group K) with attached business functions. Smart fields are designed to retrieve and manipulate specific JD Edwards EnterpriseOne table data. For example, by adding the smart field FINRPTAB - Account Balance to a report, you create a column that calculates the account balance as of a specified financial period and fiscal year.

Because the Director recognizes only the smart fields that are attached to the Report Director template, you must add business view columns to the detail section after the initial design of the report is complete. Business view fields are not available for application reports from the Director. However, the system automatically includes the Description column in the Columns in Report Section. When you are finished with the initial design of the application report, use the Business View Columns Browser in RDA to add business view columns to the report layout.

Although you cannot use smart fields in report sections without attaching them to a Director template, you can use Director templates without using smart fields.

The recommended data sequencing fields for application reports are defined in the Director template. The first two fields that are defined in the Director template become the level break fields. These fields are presented on the Data Sequencing Help form of the Director. You can modify the predefined data sequencing or override it completely while in the Director. If you select to override the data sequencing, the Director presents you with a Section Data Sequencing form. On this form, you can select from columns that are included in the attached business view.

Application reports enable you to define additional properties. Additional properties control how the report processes information and displays that information. The additional properties are defined in the JD Edwards EnterpriseOne

Director Templates application. If the Display Financial Criteria option is not selected in the Director template, the Additional Properties form is not presented by the Director in RDA.

These additional properties are available for selection in the Report Director template:

- AAI Subtotaling.

  Generates interim subtotals in the report based on the Automatic Accounting Instructions.
- Reverse Sign For.

  Reverses the signs for monetary values from how they appear in the database. For example, if you do not want to display revenues as negative values (–) and expenses as a positive values (+), as they appear in the database tables, you can select to display them both as positive values in the report section.
- Account Level of Detail Rollup.

  Indicates the level of account detail to include in the report.
- Drill Down.

  Creates a link between the data in the report and the detail in the associated application.

  See *Working with the Drill Down Feature*.
- Zero Row Suppression

  Includes the header accounts even if the detail accounts have a zero balance.

Because each application report template is unique, you do not necessarily see all of the forms as described in the tasks of this section. Some forms might not be relevant for the Report Director template that you select.

> **Note:**
> - *Enhancing Reports Using Basic Functionality*.
> - *Enhancing Reports Using Advanced Functionality*.

Understanding Smart Fields

Smart fields are reusable objects and can be used to minimize the creation of repetitive logic. Smart fields are data dictionary items with attached business functions. The business functions provide the smart field with the required logic. A smart field can have multiple business functions attached, each providing different functionality. For example, some smart fields calculate an amount and populate a column heading based on the time period that is used to calculate the amount. These smart fields must have at least two business functions attached to provide both functions.

Business functions are programs that use data structures to:

- Request specific data from JD Edwards EnterpriseOne tables.
- Return the data to the established parameters in the data structure.
- Perform calculations or other manipulation on the data.
- Send the desired information, such as column headings and complex calculations, to the report section.

Two types of business functions are available in JD Edwards EnterpriseOne:

- C business functions.

You create C business functions in the C programming language. These business functions include two parts—a source file (.c), and a header file (.h).

- Business function event rules.

  Business function event rules are also known as named event rules (NER). You create NERs using the event-rules scripting language. This scripting language is platform-independent and is stored in a database as a JD Edwards EnterpriseOne object. When you build business function event rules, the system generates C code and creates corresponding .c and .h files.

  NERs are similar to event rules that you create on a section of a report. Event rules that you create in a report are referred to as embedded event rules. While embedded event rules can be used only within the report in which they were created, NERs are created outside RDA and are reusable.

Several smart fields are shipped with JD Edwards EnterpriseOne. You can create custom smart fields to meet specific business needs. After smart fields are created, you can use them to include complex logic in report sections without having to do any programming.

For each smart field that you include in a section, you are prompted to define parameters that are specific to that smart field. The parameters are presented in the Director using a series of forms. Although the number and content of the forms vary based on the smart field, the process occurs in three phases:

- First, you are prompted to define how you want the column to appear in the section.
- Second, you are prompted to define parameters, such as period number offset, journal entry amount, and fiscal year offset.

  Some smart fields have only one parameter and, therefore, only one form in this phase. Others have multiple parameters that you specify using a series of forms.
- Third, you are prompted to filter the data that is presented in the section using data selection.

> **Note:**
> - *Working with Smart Fields*.
> - *Creating Smart Fields*.

Understanding Calculation Columns

Application reports enable you to define calculation columns. Calculation columns display the results of a calculation involving two or more smart field columns or other calculation columns. In a report that is created using a Director template, the Director recognizes only the smart fields that are attached to the template; therefore, business view fields that are included in the section layout cannot be used in calculations. After you complete the design process using the Director, you can use RDA to add calculation columns based on other columns in the report.

> **Note:** Values in calculation columns are based on the amounts that appear in the report, not on actual values that are stored in the database.

These calculation types are available when you are creating calculation columns:

- Difference between

Presents two operand fields. Use the field drop-down list boxes to select the columns to use for calculating the difference between columns.

- Percent variance between

  Presents two operand fields. Use the field drop-down list boxes to select the columns to use for calculating the percent variance between columns.
- Undefined

  Creates a numeric variable column in the report section for which you can manually define a calculation. Select the column variable in RDA, and select Define Calculation from the Column menu.
- Total of

  Presents columns from the report layout. Click each field to be included in the total.
- Product of

  Presents two operand fields. Use the field drop-down list boxes to select the columns to use for calculating the product of columns.

> **Note:**
> - *Creating Calculation Columns*.

Prerequisite

Before you can select an application report template, you must create a batch application object. The system automatically opens the welcome form of the Director as the last step in creating a batch application object.

See *Creating Report Objects*.

Selecting Application Report Templates

Access the Welcome to the Report Design Director form.

1. Select from the available header and footer options.

   Typically, you include a page header section as a standard element of a report. All other header and footer sections are optional.
2. Under the Application Reports heading, select the Director template that is most appropriate for the type of report that you are creating, and click Next.

   > **Note:** If you do not include a page header in the report, the Business View Selection Option form appears, and you can bypass the rest of the steps in this task.
3. On the Page Header Details form, select these options to allow the system to automatically populate the page header section, and then click Next:

   - Automatically add the default informational fields shown below to my page header section.

- ○ Automatically add the default informational field "Page n of Total" shown below to my page header section.

  Select this option to include the Page n of Total field in the page header. You must also select the first option if you want to include all fields in the page header.

Selecting Business Views in Application Reports

Click Next on the Welcome to the Report Design Director form to access the Business View Selection Option form.

1. Select the option I'll use the pre-defined business view.

   This option is available only when you use a Report Director template to create the report template. The predefined business view is defined in JD Edwards EnterpriseOne Report Director Templates for the specific Director template that is selected.
2. You can select one of the other options on the Business View Selection Options form to select a business view other than the predefined business view.

   > **CAUTION:** The business functions that are attached to smart fields require specific fields to process the logic. If the alternate business view that you select does not include the fields that are required by the smart fields that are attached to the Director template that you selected, the smart fields will not function.
3. Click Next.

Selecting Smart Fields in Application Reports

Click Next on the Business View Selection Option form to access the Select Columns form.

1. Drag the appropriate smart field from the Available Smart Fields list to the Columns in Report Section.

   The Smart Field Name form appears and marks the first phase for defining data for the smart field.
2. On the Smart Field Name form, enter a unique name in the Variable Name field.

   Use a name that you can easily identify when using the field in event rules or calculations.
3. Enter a descriptive name in the Report Column Headings fields.

   These fields appear on the report as the column headings unless the Smart Column Heading option is selected.

4. Select from these options, and click Next:
    - Smart Column Heading.

      Select if you want the system to use the smart field to populate the column heading. This feature enables you to create a report with column headings that change based on a rolling time period. The smart field must have a business function attached that provides this functionality.
    - Show Multiples Parameters per Page.

      Select if you want the Director to combine multiple smart field parameters on a single page. The system then displays five smart field parameters on each page of the Director for easier viewing.

      > **Note:** The smart field parameters forms vary, depending on which smart field you select. For example, if you select the FINRPTAB smart field, the parameters include the Period Number Offset, Fiscal Year Offset, and Data Selection forms.
5. On the Smart Field Parameters - Period Number Offset form, enter a value in the Enter the Literal field, and click Next.

   The text in the center of the form describes the values that are appropriate for this field. Leave the field blank to display information for the current period in the report. You can also enter a specific period number or a number that is an offset to the current period number. For example, enter **6** to display data for period six. Enter an offset of **–1** to display data for the month prior to the current month.
6. On the Smart Field Parameters - Fiscal Year Offset form, enter a value in the Enter the Literal field, and click Next.

   The text in the center of the form describes the values that are appropriate for this field. Leave the field blank to display information for the current fiscal year. You can also enter the last two digits of a specific fiscal year, such as **05**; or enter a number that is an offset to the current fiscal year, such as **+1** or **–1.**
7. On the Smart Field Parameters - Column Heading Offset form, enter a value in the Enter the Literal field and click Next.

   This form appears only if you selected the Smart Column Heading option on the Smart Field Name form. The text in the center of the form describes the values that are appropriate for this field. The value that you enter on this form should match the value that you entered on the Smart Field Parameters - Period Number Offset form. This ensures that the column heading reflects the data that is included in the column.
8. On the Smart Field Data Selection form, complete the fields and click Finish.

   These fields define the data that you want to appear in the smart field column. The fields are defined in the smart field template and vary depending on the Report Director template that is selected.
9. If the fields that appear on the Smart Field Data Selection form do not reflect the data selection criteria that is required, select the Advanced option, define data selection manually, and then click Finish.
10. To review or revise smart field parameters, click a selected smart field on the Select Columns form, and then click the Define Smartfield button.

Creating Calculation Columns in Application Reports

Access the Select Columns form.

1. Click the Define Calculation button.
2. On the Define Calculation form, enter a descriptive name in the Calculation Name field.

   This name appears on the report as the column heading. Use names such as **Total of Account.**

3. Select a calculation type.
4. Define the operands for the calculation type that you selected, and click Finish.

   Use the drop-down list box to select from valid smart field columns.

Defining Section Data Sequencing in Application Reports

Click Next on the Select Columns form to access the Data Sequencing Help form.

1. Sequence the section data using the predefined fields from the Report Grouping list.

   You can clear the predefined fields as appropriate.
2. Further sequence the data using predefined fields from the Report Detail list, and then click Next.

Defining Section Data Sequencing Using the Advanced Option

Access the Data Sequencing Help form.

1. If you want to add to the predefined data sequencing, select the Advanced option to *append* additional data sequencing, and then click Next.
2. If the fields that appear on the Data Sequencing Help form do not reflect the data sequencing that is required, clear the predefined fields and select the Advanced option to *override* the predefined data sequencing, and then click Next.
3. On the Section Data Sequencing form, select business view columns from the Available Columns list, and move them to Selected Columns.
4. When the section data sequencing is complete, click Next.
5. Define sort order, level breaks, and page breaks on the Define Sort Properties form, and click Next.

Including Records in Application Reports

Click Next on the Define Sort Properties form to access the Help with Section Data Selection form.

1. To include data from the balance sheet accounts only, select the Select only BALANCE SHEET accounts from the automatic accounting instructions option.

   This option includes data fields that are established in the Oracle's JD Edwards EnterpriseOne General Purpose (GLGxx) automatic accounting instructions (AAIs) as balance sheet accounts.
2. To include data from the income statement accounts only, select the Select only INCOME STATEMENT accounts from the automatic accounting instructions option.

   This option includes data fields that are established in the JD Edwards EnterpriseOne General Purpose (GLGxx) AAIs as income statement accounts.
3. To *append* data selection to the default income statement or balance-sheet data selection, select the I'll add my own data selection to the above balance sheet or income statement criteria option.

   The system provides the criteria for including the income statement and balance sheet accounts. You can define additional criteria.
4. To define data selection manually, select the Set up data selection manually option.
5. On the Data Selection form, define data selection and click Next.

Defining Additional Properties in Application Reports

Click Next on the Help with Section Data Selection form or the Section Data Selection form to access the Additional Properties form.

1. Select from the additional properties that are presented.

   The properties that are available depend on the Director template that you selected. The properties are defined in the Director template
2. Click Next.

Creating Batch Versions of Application Reports from the Director

Click Next on the Additional Properties form to access the Finish form of the Director.

1. To allow the system to automatically generate a batch version of the report template, select the Yes, create a version of this report option.
2. Enter the name of the version in the field beneath the Yes, create a version of this report option.
3. To review the selections, click Back to move backwards through the Director forms.

   On the Navigation Assistant, you can also click the name of the form that you want to review.
4. When you are satisfied, click Finish.

> **Note:** When you click Finish, you can no longer access the Director for this report. Before you click Finish, you can review the selections on all forms of the Director.

Saving and Reviewing Reports

This section provides overviews of how to save reports, how to review the report design, and how to preview reports, lists the prerequisites, and discusses how to:

- Save reports
- Preview reports

Understanding How to Save Reports

When you click Finish on the Director, the initial design process for the report template is complete. You cannot return to the Director after you click Finish. All modifications to the report template must be performed manually using the features of RDA.

You cannot save the report template while using the Director. You should save the report template in RDA immediately after completing the initial design.

Understanding How to Review the Report Design

After you complete the report design process using the Director, RDA displays the report template based on the selections that you made. The Report tab displays the sections of the report template and the fields that you selected for the layout. Each report section includes:

- An icon that indicates that the section type appears on the section tile.
- If you defined data selection for the section, a funnel-shaped icon on the section tile.
- A title description on the section tile.

  The title reflects the name of the business view that is attached to the report section. The title of the main detail section of the report is in a bold font to distinguish it from the other report sections.
- Corner brackets around each field within the section.

Understanding How to Preview Reports

Use the Preview tab to view how the report will appear when you process it. From the Preview tab, you can review the content of the report and the format. Return to the Preview tab as you make design changes. When you select the Preview tab, the system prompts you to save the report before previewing.

> **Note:** Each time that you return to the Preview tab, you must refresh the view so that the report reflects the modifications that you made.

RDA uses Adobe Reader to present the report preview. The Preview tab includes an Adobe Reader toolbar. You can use all of the functions that are available on the toolbar to view the report.

The purpose of the preview feature is to review the layout, the results of attached logic, and a sampling of the data. Preview does not necessarily present all records that will be fetched from the database when the report is processed. You can define the number of rows to preview from User Options.

See *Configuring the Report Design Aid Workspace*.

Prerequisites

Before you can save reports, you must:

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

Saving Reports

Open a report in RDA.

Perform one of these actions to save the report template:

- From the File menu, click Save.
- Click the Save button on the toolbar.

Previewing Reports

Open a report in RDA.

1. Select the Preview tab.

   The Report Preview form appears and prompts you to run the preview.
2. Click Yes on Report Preview.
3. Click Yes to save the report.
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.

5 Configuring the Report Design Aid Workspace

Understanding the RDA Design Workspace

You can modify the design workspace in RDA to accommodate your personal work style. You can maximize the workspace by displaying many of the elements only when they are needed. You can modify the tightness of the grid to enable you to easily align fields. You can display rulers to help you size the report template to properly fit on a page.

RDA offers a variety of ways to configure the design workspace:

- Set user options.

  Enables you to display or hide elements of the design workspace and set the number of rows to appear in Preview.
- Set grid alignment.

  Enables you to tighten up, or loosen up, the grid for aligning fields.
- Show and hide the display tree.

  Enables you to view the sections and fields of the report template in a hierarchical tree structure.
- Show and hide the Business View Columns Browser.

  Enables you to view and select business view columns from the business view that is attached to the selected detail section. The Business View Columns Browser is a window that you can either dock or float over the design workspace.
- Show and hide the Data Dictionary Browser.

  Enables you to search for and select data dictionary items to include in the layout of the selected section. A window is available that you can either dock or float over the design workspace.

Configuring the Design Workspace

This section provides overviews of user options, grid alignment, and the display tree, lists the prerequisites, and discusses how to:

- Set user options.
- Set the grid alignment.
- Show and hide the display tree.
- Show and hide the Business View Columns Browser.
- Show and hide the Data Dictionary Browser.

Understanding User Options

You can set user options to show or hide the elements of the user interface. You can also control the number of rows to appear on the Preview tab. User options are also referred to as report view options. Three categories of user options are available:

- General
- Rulers
- Preview

General user options affect the appearance of elements in the design workspace. General user options include:

- Show Invisible Sections at Startup.

  Displays sections that are defined as invisible when you open the report template. You must have sections of the report defined as invisible for this option to affect the RDA view.
- Show Section Titles.

  Displays the section title and related icons in the tile to the left of each section.
- Show Right Margin.

  Places diagonal lines outside the right margin in the design workspace. The lines indicate that fields reside outside of the right-page margin. This feature is helpful when you change the orientation of the report, such as changing from landscape to portrait.
- Show Tabs.

  Displays the Report, Preview, Attachments, and other section-specific tabs at the top of the design workspace.
- Show Navigational Assistant.

  Displays the Navigation Assistant in companionship with the Director. Hiding the Navigation Assistant in User Options hides it for all future design sessions.
- Show Data Dictionary Text Overrides.

  Displays a small green triangle in the bottom-right corner of column headings when the name appears differently on the report than in the data dictionary. The indicator appears in the design workspace only; it does not print on the report.
- Allow Smart Field Template Selection.

  Enables you to select smart field templates from the General tab of the appropriate section properties form.
- Enable System Language Font Override and Reposition.

  Dynamically repositions fields based on font changes that affect the field size. This option is beneficial for global reporting when you must present data in different languages with fonts of different widths and heights. This option is disabled by default.
- Show CSV Tip Dialog.

  Displays the CSV Tip dialog box, which indicates the spacing recommendation for exporting reports to comma separated value (CSV) files.

- Show Object Count Warning after Row Generation.

  Displays a warning message that indicates when the report has exceeded the recommended size. This is typically an issue with reports that use the automatic row generation feature.

Understanding Grid Alignment

The grid alignment feature enables you to tighten or loosen the design grid for flexibility in positioning fields in a group section. The closer the dots, the tighter the grid. A tighter grid enables you to move fields in smaller increments for more precise positioning.

The vertical spacing value represents pixels on the workstation that are used for designing the report. The value is converted to a workstation-independent measurement when it is saved. Set this option to ensure that the report maintains the same proportions when it is viewed on a different workstation.

The following options are available for grid alignment:

- Horizontal spacing.

  Adjusts the horizontal grid spacing for the entire report template. If you designed the report to be exported to a CSV file, set the horizontal spacing to 52. This value corresponds to the default column width in Microsoft Excel.
- Vertical spacing.

  Adjusts the vertical grid spacing for the entire report template.
- Display the grid.

  Displays a grid of dots to assist in positioning fields. When this option is cleared, no grid appears in the design workspace.
- Snap to grid.

  Aligns fields with the nearest grid line intersection even if the grid is not displayed. When this option is cleared, you can position fields that do not line up with the grid. If you designed the report to be exported to a CSV file, the Snap to Grid option should be selected.

Understanding the Display Tree

The display tree presents the sections and fields of the report in a hierarchical tree structure. You can show the display tree and then either dock it or float it over the design workspace. The display tree presents the sections that are included in the report, as well as the objects that are associated with each section, including invisible fields. The highest component of the tree indicates the name of the report. Under the report name, the system displays the sections of the report in the order in which they were created.

When you expand a section, the system displays a Fields folder. Expand this folder to view a list of objects that are included in the section. Tabular rows and cells do not appear in the display tree. If you expand the Fields folder of a detail section, two entries appear for each field; one entry represents the constant (header portion of the field) and the other represents the variable (data portion of the field).

> **Note:** The display tree presents the structure of the report only; it does not represent the processing flow of the report. Use the Preview tab to view the format of the report.

Prerequisites

Before you can configure the design workspace, you must:

- Create a batch application object.
- Ensure that a business view is attached to the detail section of the report template.

Setting User Options

Open a report in RDA.

1. From the View menu, select User Options.
2. On the User Options form, select from the available options under the General heading, and click Apply.

   Apply enables you to view modifications without quitting the form.
3. Select the Show Rulers option, and then select from these options in the Ruler Units field:
   - Inches
   - Centimeters
   - Points
4. Select the Prompt Before Running Preview option to present a prompt before running the preview.
5. Enter the number of rows to process in the Rows to Preview field, and click OK.

   Enter a number that provides enough records to enable you to verify the format of the report and review totaling or other logic, while keeping the number of rows that are processed to a minimum.

Setting the Grid Alignment

Open a report in RDA.

1. From the Layout menu, select Grid Alignment.
2. On the Alignment Grid form, modify these spacing options as required:
   - Horizontal
   - Vertical
3. Select from these options, and click OK:
   - Display Grid
   - Snap to Grid

Showing and Hiding the Display Tree

Open a report in RDA.

1. From the View menu, select Report Tree View.
2. Click the + or – sign to expand (+) and collapse (–) the display tree.
3. Double-click an object in the tree structure to view or change object properties.

Showing and Hiding the Business View Columns Browser

Open a report in RDA.

1. Click a detail section in the report.
2. From the View menu, select Business View Columns Browser.

   > **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.

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

Understanding Properties

Properties are available in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) at the report, section, and object levels. Properties enable you to:

- Change fonts.
- Change field justifications.
- Change decimals.
- Select additional advanced settings.

Some properties are available at all levels, while other properties are specific to a level. Properties that are available at all levels are processed by the system using a hierarchy. When properties are defined at multiple levels, the lowest level properties are recognized and processed by the system. For example, if you define a font size of 8 in the report properties, then define a font size of 10 in the properties of an individual field, the individual field appears in a font size of 10, while the rest of the report appears in a font size of 8.

This table describes how properties affect the elements of a report template:

Properties	Description
Report properties	Affect the entire report.
Section properties	Affect the selected section only. You can view and modify section properties using either the menu method or the double-click method.
Field properties	Affect the selected field only. You can view and modify data field properties using the menu method, the Fields tab on the report section method, or the double-click method.
Column heading properties	Affect the selected column heading only. You can view and modify column heading properties using the menu method, the Fields tab on the report section method, the Column tab on the design workspace, or the double-click method.
Row properties	Affect the selected row only. These properties exist in tabular sections only. You can view and modify row properties using the menu method, the Row tab on report section method, the Row tab on the design workspace, or the double-click method.

Modifying Properties

This section provides overviews of report properties, section properties, and field properties, lists the prerequisites, and discusses how to:

- Modify report properties.
- Modify section properties.
- Modify field properties.
- Modify column properties.
- Change column heading names.
- Modify column properties from the Column tab.
- Modify row properties.

Understanding Report Properties

Report properties are accessed from the File menu in RDA. Report properties affect the entire report and include these tabs:

Properties Tab	Description
Report Properties	Enable you to select totaling, output, and performance options.
Font/Color	Enables you to modify fonts, font styles, font size, and font colors for the entire report.
Cover Page Options	Enable you to select the cover page option: select general and section options to include on the cover page.
Decimal Scaling	Enables you to apply decimal scaling to numeric fields. This option affects only tabular section reports.
Advanced	Enables you to select subsystem jobs, paper size, dynamic positioning, and transaction processing, and to override environments options.

When making report level changes, you must consider two scenarios:

- You design a new report template that has not yet been saved.

  You can make report level changes, save the report template for the first time, and quit RDA. If you allowed the system to create a batch version automatically, the batch version reflects the report level revisions.
- You modify an existing report template.

  Changes to properties at the report level are not reflected in existing versions of the report if the report template has already been saved and you have quit RDA prior to making the report level change. For example, you open an existing report and make report level changes. These report level changes are *not* reflected in existing versions. You must also make the modification at the version level or create a new batch version. Batch versions that are created *after* you revise the report level properties reflect the report level revisions.

Understanding Section Properties

Section properties are accessed from the Section menu in RDA. You can also right-click or double-click a report section to access the section properties.

Section properties for group and columnar sections are the same. Group and columnar section properties affect the current section only and include these tabs:

Properties Tab	Description
General	Enables you to modify the section description.
Font/Color	Enables you to modify fonts, font styles, font size, and font colors.
Fields	Enables you to access properties for the fields in the section layout and make fields invisible.
Advanced	Enables you to select Visible, Absolute Position (text wrapping), Page Break After, Conditional, and Reprint At Page Break options.

Tabular sections provide additional section properties. Tabular section properties affect the current section only and include these tabs:

Properties Tab	Description
General	Enables you to modify the section description.
Font/Color	Enables you to modify fonts, font styles, font size, and font colors.
Financial Reports	Enable you to select Drill Down, Perform AAI Subtotaling, Reverse Sign for, Zero Row Suppression, Level of Detail Rollup, and Financial Descriptions options.
Fields	Enable you to access properties for the fields in the section layout and make fields invisible.
Row List	Enables you to access row properties and make rows invisible.
Cell Overrides	Enable you to access cell override properties and make cells invisible.
Column Override	Enables you to select a section from another report from which to import columns.
Row Override	Enables you to select a section from another report from which to import rows.

Properties Tab	Description
Decimal Scaling	Enables you to apply decimal scaling to numeric fields.
Advanced	Enables you to select Visible, Absolute Position (text wrapping), and Page Break After options.

When you define fields as invisible on the Fields tab of the appropriate section properties form, a circle with a line through it appears next to the field icon. When column headings are invisible, their associated column variable is also invisible. Double-click the icon to toggle the visible or invisible property.

Fields Property Tab of the Section Properties Form

In the Visible column of the Fields tab, the various field types are represented by these icons:

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.

Properties Tab	Description
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.

Advanced Tab	Description
Visible	Enables you to toggle a field between visible and invisible.
Suppress At Totals	Enables you to suppress a field from being totaled. This is especially useful in tabular sections where numeric fields are totaled automatically by the system.
Print On Change Only	Enables you to define a field to print only when the value changes. For example, you have a report that lists several records for each of five business units. Instead of the same business unit value printing next to each associated record, the business unit value prints once for the first record. Then when the data for the next business unit begins, the new business unit value prints for the first associated record.
Global Variable	Enables you to define a field so that it can be accessed in Event Rules Design from another section of the report.

Prerequisites

Before you begin modifying properties, ensure that you:

- Create two batch application objects.
- Complete the design of one report template using columnar or tabular sections.
- Complete the design of the second report template using a tabular section with rows.

Modifying Report Properties

Open a report in RDA.

1. From the File menu, select Report Properties.
2. On the Properties form, select the Report Properties tab.
3. Review and modify totaling, output, and performance options as appropriate.
4. Select the Font/Color tab to review and modify font properties for the entire report.

   To override any font property settings of individual objects, select the Apply settings to all Objects option.

   To return the font properties to the system default, click the Defaults button.
5. Select the Cover Page tab, select the option to include a cover page with the report, and review and modify cover page options.

   Selecting the cover page in Report Properties does not automatically print the cover page with the report. You must select the cover page option on the Version Detail form of the Oracle's JD Edwards EnterpriseOne Batch Versions (P98305) application.
6. Select the Decimal Scaling tab, and review and modify decimal scaling for numeric fields in a tabular section.

   This tab applies to tabular sections only. Select the Apply settings to all Objects option to override any decimal scaling settings of individual objects.

   To return the decimal scaling properties to the system default, click the Defaults button.

   See *Defining Decimal Scaling*.
7. Select the Advanced tab to review and modify additional report properties.

Modifying Section Properties

Open a report in RDA.

1. Click the section for which you want to modify properties.
2. From the Section menu, select Section Properties.
3. On the General tab, you can change the section title and review the business view that is attached to the section.

   You might want to change the section title if you have two sections with the same title. This might be the case when you create custom sections that use the same business view as the detail section.
4. Select the Font/Color tab to review and modify font properties for the entire section.

   Select the Apply settings to all Objects option to override any font property settings of individual objects.

   To return the font properties to the system default, click the Defaults button.
5. Select the Fields tab to review the fields that are included in the section layout.

   You can select a field and click the Field Properties button to modify the field properties of the field.

   Double-click the icon of a field to toggle between invisible and visible.
6. Select the Advanced tab to review and modify additional section properties.

The advanced options vary depending on the selected section type.

Modifying Field Properties

Open a report in RDA.

1. Click the variable for which you want to modify properties.
2. From the Edit menu, select Item Properties.
3. On the Description tab, override the name of the field as it appears in Event Rules Design.
4. Select the Font/Color tab to review and modify font properties for the field.

   To return the font properties to the system default, click the Defaults button.
5. Select the Style tab to review and modify lines and rectangles.
6. Select the Display tab to review and modify the justification of the field

   To return the display properties to the system default, click the Defaults button.
7. Select the Totaling tab to review and modify total and grand total options.
8. Select the Advanced tab to review and modify additional field properties.

Modifying Column Properties

Open a tabular or columnar section report in RDA.

1. Click the column heading for which you want to modify properties.
2. From the Edit menu, select Item Properties.
3. On the General tab, override the name of the column heading.

   If you are using the column heading in event rules, modify the variable name so that the column is easy to recognize in Event Rules Design.
4. Select the Font/Color tab to review and modify font properties for the column heading.

   To return the font properties to the system default, click the Defaults button.
5. Select the Style tab to review and modify lines and rectangles.
6. Select the Advanced tab to toggle the column between visible and invisible.

   When you make the column heading invisible, the column variable is also invisible.

Changing Column Heading Names

Open a columnar or tabular section report in RDA.

1. Click the column heading for which you want to change the name.
2. From the Edit menu, select Item Properties.
3. On the Column Heading Properties form, select Change Name if the column is from the business view.

   If the column is not from the business view, go to the next step.
4. Enter a new descriptive name in the Variable Name field.

The variable name of a column heading appears in event rules with the prefix of RC (Report Constant). Use this option if you need to attach logic to the column heading.

5. Select Override Col Headings (Override Column Headings) if the column is from the business view.

   If the column is not from the business view, go to the next step.
6. Change the text in the Col Heading 1 (Column Heading 1) and Col Heading 2 (Column Heading 2) fields, and click OK.

   > **Note:** You should always change the variable name of a data field to match its column name. This makes managing the data field easier, especially if you plan to attach event rules to the data field.
7. To return the column heading names back to the default, clear the Change Name and Override Col Headings options.

Modifying Column Properties from the Column Tab

Open a tabular or columnar section report in RDA.

1. Click the detail section for which you want to modify column properties.
2. Select the Column tab at the top of the design workspace.
3. To modify a property, double-click the property for a column and access the appropriate control.
4. To determine which fields appear on the Column tab, right-click anywhere on the row or column headers of the detail area.
5. Click Field Selection.
6. On the Object Design Properties form, select or clear any of the available properties to show or hide them on the Column tab, and click OK.

   Select the Basic and Advanced tabs on the Object Design Properties form to view all available properties.

Modifying Row Properties

Open a tabular row report in RDA.

1. Click the row for which you want to modify properties.
2. From the Edit menu, select Item Properties.
3. On the General tab, review and modify the name and description of the row.
4. Select the Font/Color tab to review and modify font properties for the row.

   To return the font properties to the system default, click the Defaults button.
5. Select the Style tab to review and modify lines and rectangles.
6. Select the Display tab to review and modify the justification.

   To return the display properties to the system default, click the Defaults button.
7. Select the Options tab to review and modify the space before the row.
8. Select the Cell Overrides tab to review the cell overrides that are defined in the row.

   You can select a cell override and click the Cell Override Properties button to modify the properties of the selected cell.
9. Select the Advanced tab to review and modify additional row properties.

See *Working with Rows*.

See *Overriding Cells and Cell Properties*.

7 Working with Objects in Report Sections

Understanding Report Objects

Typically, you include multiple sections in a report template; you usually include a page header and one detail section. In each report section, you can include multiple report objects. These objects are either business view fields or other data fields.

Some report objects are available only in specific section types. For example, you typically use runtime fields in page header sections; runtime fields are not available in columnar and tabular sections. Business view fields are used in detail sections only. You cannot attach a business view to a page header. In contrast, you can place data dictionary fields in any section type.

Working with Report Objects

This section provides overviews of business view columns, data dictionary fields, and data fields, lists the prerequisites, and discusses how to:

- Add and remove business view columns.
- Add and remove data fields.
- Change data field names.
- Add and remove data dictionary fields.
- Disconnect constants from variables in group sections.
- Perform in-section totaling.

Understanding Business View Columns

You can add or remove business view columns from any detail section that has a business view attached. Business views are subsets of columns that reside in one or more tables. The columns that are included in a business view were selected to meet specific business needs.

When you are adding business view columns or other data fields to *columnar* and *tabular* sections, the new column is inserted to the *right* of any column on which you place the cursor. When you do not place the cursor on a column, the inserted column is inserted to the *left* of the first column, becoming the first column in the section. You can then drag the new column to another location in the same section. You cannot move the column to another section of the report. If you have placed the field in the wrong section, you must delete the field and reenter it into the correct section.

After selecting a business view column or other data field for a *group* section, you must click in the group section to position the data field. You can position the new field anywhere in the group section because of the free-form layout.

Understanding Data Dictionary Fields

Data dictionary fields are populated using event rules. Business view columns fetch data from the database, but data dictionary fields, although they have specifications attached, do not have data associated with them. Data dictionary fields are defined in the data dictionary and can have edit codes and special triggers attached. Data dictionary fields are also referred to as data items.

To add a data dictionary field to a report section, click the report section and use the Data Dictionary Browser to search for and select an appropriate data dictionary field. Typically, you want to select a data dictionary field that includes the specifications that are required by the report. For example, you are adding a field to a salary report to calculate raises. Use the Data Dictionary Browser to select the Rate-Salary, Annual (SAL) data dictionary field so that the raise amount is in the same format as the salary amount.

You must add logic to the data dictionary field for the field to present data. You can click the report section where the data dictionary field resides and enter Event Rules Design. You must select an event on which to create the logic. Use the features of Event Rules Design to create logic that populates the data dictionary field with the correct data. For example, to populate the raise amount data dictionary field, you might select the Do Section event, create an assignment to assign a value to the data dictionary field, and create an expression that multiplies the annual salary by a percentage to calculate the raise amount.

See *Working with Event Rules*.

Understanding Data Fields

Data fields are individual data containers that reside in report sections. Page numbers, dates, and report names are all examples of data fields. You can add data fields to any type of report section, although you cannot add every data field type to every report section type. For example, you typically add data fields (such as report name, date, and so forth) to the page header of a report.

After creating the report, you can:

- Insert data fields.
- Delete data fields.
- Change data field names.
- Change column heading text.
- Disconnect a constant from its variable.

  In a group section, you can separate the constant text from the variable that it describes. For example, you can modify a report that contains the Business Unit field and its description by changing the constant to read **Business Unit Number and Name.** You can then disconnect the Description text from the Description variable, delete the text, and still retrieve the value from the JD Edwards EnterpriseOne table.

- Perform in-section totaling.

  When you are not concerned with how calculations are presented, you can include calculations within columnar and group sections. For example, you can perform calculations in one hidden section for use in another. You can calculate a total, a grand total, or both.

  When you want to format and display totals in columnar and group section reports, perform the calculations in level break footers. You can provide descriptive labels for the aggregates in level break footers.

  See *Working with Level Break Sections*.

This table describes the data fields that are available in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA):

Field	Description
Constants	Static fields that display a string of text, such as a company name inserted in the page header. You can insert constant fields into any report section.
Alpha Variables	Fields that contain alphanumeric data that are populated using an event rule. You can insert alpha variables into any report section.
Numeric Variables	Fields that contain numbers and are typically used for calculations. You can insert numeric variables into any report section.
Date Variables	Fields that contain dates. You can insert date variables into any report section.
Report Date	Runtime field that contain the date when the report is run. Report dates are typically used in page headers, but you can insert them into any report section except columnar and tabular.
Report Time	Runtime field that contain the time at which the report is run. Report times are typically used in page headers, but you can insert them into any report section except columnar and tabular.
Page Number	Runtime field that display the current page number. Page numbers are typically used in page headers, but you can insert them into any report section except columnar and tabular.
Page n of Total	Runtime field that display both the current page number and the total number of pages in the report (such as Page 4 of 10). Page n of Total fields are typically used in page headers, but you can insert them into page footers.
Company Title	Runtime field that contain the name of the default company (company 00000). Company titles are typically used in page headers, but you can insert them into any report section except columnar and tabular.
Report Title	Runtime field that contain the report title. Report titles are typically used in page headers, but you can insert them into any report section except columnar and tabular.

Data fields that you insert into columnar and tabular sections appear in columnar format. Column headings are constants with associated variables. Column headings and variables are attached, so when you move or delete one, RDA moves or deletes the other.

In group sections, data fields also comprise constants and variables, but they appear side-by-side instead of in a columnar format. Unlike columnar and tabular sections, you can move constants and variables independently of each other. Furthermore, by disconnecting the two, you can separate the constant from the variable and delete one without deleting the other. When you delete the constant, you retain the variable, and the associated data still appears in the report.

You must click the intended report section prior to selecting a data field. If you place the data field in the wrong section, you must delete the field and reenter it into the correct section. You cannot move the data field to another section of the report. If you selected the wrong section in a group section report and notice after you select the data field, you must drop the data field in the incorrect section and then delete it. You cannot select another data field until you position the one that you have already selected.

You can modify the appearance of data fields by changing the headings, moving them, changing the size, font, or color, or associating them with lines or boxes. You can modify the behavior of data fields by attaching event rules.

Prerequisites

Before you begin working with report objects, ensure that you:

- Create two batch application objects.
- Complete the design of one report template using a group section.
- Complete the design on the second report template using a columnar or tabular section.

Adding and Removing Business View Columns

Open a report in RDA.

1. Click a detail section with an attached business view.
2. From the View menu, select Business View Columns Browser.

   This form displays columns from the business view that is attached to the selected detail section.
3. Drag one or more columns into the detail section.

   If the selected section is a group section, you must click the section to position the field.
4. From the View menu, select Business View Columns Browser to close the browser.
5. Drag the inserted column to the appropriate location.
6. To remove business view columns, select the column (either the constant or the variable), and select Delete from the Edit menu.

Adding and Removing Data Fields

Open a report in RDA.

1. Click the report section to which you want to add a data field.
2. From the Insert menu, select the data field that you want to add.

   The data field selection varies, based on the selected report section.
3. Drag the selected data field to the selected section.

If the selected section is a group section, you must click the section to position the field.

4. Drag the inserted data field to the appropriate location.
5. To remove data fields, select the field (either the constant or the variable), and select Delete from the Edit menu.

Changing Data Field Names

Open a report in RDA.

1. Click the variable for which you want to change the name.
2. From the Edit menu, select Item Properties.
3. On the appropriate properties form, enter a descriptive name in the Variable Name field.

> **Note:** When you change the variable name of a constant field in any section, except for columnar and tabular sections, the change is reflected on the report. If you change the variable name of a constant field, you should change the name of the associated variable to make managing the data field easier, especially if you plan to attach event rules to the data field. All other variable name changes do not affect the appearance of the data field directly.

4. If the data field is a column heading, on the Column Heading Properties form, change the text in the Col Heading 1 (Column Heading 1) and Col Heading 2 (Column Heading 2) fields, and click OK.

Adding and Removing Data Dictionary Fields

Open a report in RDA.

1. Click the report section to which you want to add a data dictionary field.
2. From the View menu, select Data Dictionary Browser.
3. Use the QBE (query by example) line to search for and select an appropriate data dictionary field.
4. Drag the data dictionary field to the report section.
5. From the View menu, select Data Dictionary Browser to close the browser.
6. Drag the inserted data dictionary field to the appropriate location.
7. To remove data dictionary fields, select the field, either the constant or the variable, and select Delete from the Edit menu.

Disconnecting Constants from Variables in Group Sections

Open a group section report in RDA.

1. In the group section, click *either* the constant or the variable of the field that you want to modify.
2. From the Edit menu, select Disconnect.

> **Note:** After you disconnect the constant from its variable, no option is available to reconnect them on the report. From the Edit menu, you can select Undo to undo *one action only.* If necessary, you can delete the disconnected text and variable and reinsert the business view column. The new field appears as it did originally with a constant field that is linked to its variable.

3. Click *either* the constant or the variable of the field, whichever one you want to delete, and select Delete from the Edit menu.

Performing In-Section Totaling

Open a columnar or group section report in RDA.

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.

8 Working with Objects Unique to Tabular Sections

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.

Working with Row Description Columns

This section provides an overview of Row Description columns, lists the prerequisites, and discusses how to:

- Create Row Description columns manually.
- Delete Row Description columns.

Understanding Row Description Columns

When you create tabular sections, the system automatically includes a Description column in the section layout. The data in this column is based on the data sequencing fields that you define as level break fields. If you remove the Description column from the section layout, you can select it from the list of business view columns.

If the level break fields that are included in the Description column have data dictionary triggers attached, the system prints the description of the field in the Row Description column. Otherwise, the system prints the field value. For example, if you define company as a level break field and retrieve records for company 00001, the description of Financial Reporting Company appears in the Row Description column. This is because the company field has a trigger attached in data dictionary. If no trigger were attached to the company data dictionary field, the system would display the field value of 00001.

The Row Description column has special capability for level breaks that are associated with the Subledger, Cost Object, and Object Subsidiary fields. When a level break causes a change in one of these fields in a tabular section, the Description row automatically displays this information:

- Subledger and subledger type.
- Cost object and cost object type.
- Object subsidiary and object subsidiary type.

Prerequisites

Before you begin working with Row Description columns, ensure that you:

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

Creating Row Description Columns Manually

Open a tabular section report in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

1. Click the tabular section in which you want to insert a Row Description column.
2. From the Column menu, select Create, Row Description Column.

   The Description column appears.
3. Drag the column to the appropriate location.

   The Row Description column is typically the first column in tabular sections.

Deleting Row Description Columns

Open a tabular section report in RDA that contains Row Description columns.

1. Click the Description column.
2. From the Edit menu, select Delete.

Defining Decimal Scaling

This section provides an overview of decimal scaling, lists the prerequisites, and discusses how to:

- Change decimal scaling for individual fields.
- Change decimal scaling for all fields in a tabular section.
- Change decimal scaling for all fields in all tabular sections of a report.

Understanding Decimal Scaling

Decimal scaling enables you to simplify the way large numbers appear in tabular sections. You can change decimal scaling for a single field in a tabular section, for all fields in a tabular section, or for all fields in all tabular sections of a report.

This table illustrates how the amounts in a report appear when the numeric data is scaled to 1000:

Numeric Data	Displayed Value
100,000.42	100
10,041.62	10
1,021.75	1
1,512.69	1.5

When modifying decimal scaling at the report level, you must consider the existing batch versions.

See *Working with Report, Section, Field, Column, and Row Properties*.

Prerequisites

Before you begin working with objects unique to tabular sections, ensure that you:

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

Changing Decimal Scaling for Individual Fields

Open a tabular section report in RDA.

1. In the tabular section, double-click the variable of the numeric column that you want to change.

2. On the Column Variable Properties form, select the Decimal Scaling tab.
3. Select the appropriate level of decimal scaling, and click OK.

   To return the field to its default decimal scaling setting, click the Defaults button.

Changing Decimal Scaling for All Fields in a Tabular Section

Open a tabular section report in RDA.

1. Double-click the tabular section that you want to change.
2. On the Section form, select the Decimal Scaling tab.
3. Select the appropriate level of decimal scaling, and click OK.

   To return all of the fields in the section to their default decimal scaling settings, click the Defaults button.

   The decimal scaling changes that you make on this form affect all of the numeric fields in the section, except for those fields that have been modified individually. To override individual settings and apply the changes to all fields in the section without exception, select the Apply settings to all Objects option.

Changing Decimal Scaling for All Fields in All Tabular Sections of a Report

Open a tabular section report in RDA.

1. From the File menu, select Report Properties.
2. On the Properties form, select the Decimal Scaling tab.
3. Select the appropriate level of decimal scaling, and click OK.

   To return all of the fields in all tabular sections of the report to their default decimal scaling settings, click the Defaults button.

   The decimal scaling changes that you make on this form affect all of the numeric fields in all tabular sections of the report. The exceptions are all fields that are located in a tabular section where the section decimal scaling has been defined and fields that have been modified individually. To override these settings and apply the changes to all of the fields in all tabular sections of the report without exception, select the Apply settings to all Objects option.

   > **Note:** When modifying decimal scaling at the report level, you must consider the existing batch versions.

   See *Working with Report, Section, Field, Column, and Row Properties*.

Creating Calculation Columns

This section provides an overview of calculation columns, lists the prerequisites, and discusses how to:

- Define calculation columns
- Define percent calculations

- Remove column calculations

Understanding Calculation Columns

Calculation columns contain the results of mathematical calculations. You can perform a calculation involving any number of columns.

When you create an application report using the Director, you have the option on the Select Columns form to create calculation columns. From the Director, you can create calculation columns using only the smart field columns that you selected for the section layout.

After the initial design of the report template is complete, you can insert calculation columns. You can create these calculation columns using:

- Any column from the attached business view.
- Smart field columns that are included in the section layout.
- Other calculation columns.

> **Note:** Calculation columns are based on the amount signs (debit or credit) as they appear in the report, not the actual value that is stored in the database.

Percent Calculations

You can display numbers in a column as a percent of the total in another column. This type of calculation is used in all reporting types, but most often in financial reports such as standard income statements. When used in income statements, the percent calculation is referred to as the percent of revenue.

Prerequisites

Before you begin working with calculation columns, ensure that you:

- Create a batch application object.
- Complete the design of the report template using either a tabular section or an application report.
- Perform these actions:
  - Create a percent calculation column from the Director.

    For example, if you are creating a financial report that is based on account balances, create a column called Percent of Revenue.
  - Insert a numeric variable column.
  - Add rows to the tabular section.

    This step is necessary to create the cell to use as the denominator. For example, add data rows to hold the revenue sales and a calculation row to hold the total revenue.

Defining Calculation Columns

Open an application report or a tabular section report in RDA.

1. Select the tabular section in which you need to add calculation columns.
2. Insert a data dictionary field or a numeric variable to hold the calculated value.
3. Drag the field to the appropriate location.
4. Change the name of the column heading and variable to reflect the purpose of the calculation.

   See *Changing Column Heading Names*.
5. Click the newly created column.
6. From the Column menu, select Define Calculation.
7. On the Expression Manager form, define the calculation by performing these actions, and click OK:
   a. Double-click fields from the Available Information list.
   b. Click the appropriate calculator functions to build the expression.

Defining Percent Calculations

Open an application report or a tabular section report in RDA.

1. After adding the appropriate rows to the tabular section, click the column that you inserted for the percent calculation.
2. From the Column menu, select Define Calculation.
3. On the Expression Manager form, define the percent calculation and click OK.

Removing Column Calculations

Open a tabular section report that contains calculation columns in RDA.

1. Click the column variable for which you want to remove the calculation.
2. From the Column menu, select Remove Calculation.
3. Create a new calculation on the column or delete the column.

Working with Rows

This section provides an overview of tabular rows, lists the prerequisites, and discusses how to:

- Add data rows
- Add calculation rows
- Add sum rows
- Add underline rows
- Add constant rows

- Generate rows automatically

Understanding Tabular Rows

In tabular sections, rows consist of information that is presented horizontally. Typically, rows contain data that is read from individual database records; however, you can add rows to include details in the report section such as:

- Underlines
- Blank lines
- Special calculations

Tabular sections are the only detail sections for which you can define rows. In all cases, after you create a row, you can drag it to a new location or delete it by selecting Delete from the Edit menu.

Rows can be defined manually or through automatic row generation. Automatic row generation is used primarily in financial reports. A report that displays a chart of accounts is an example of data that can be generated using the automatic row generation feature.

Rows that are defined manually are defined individually and each data row includes its own data selection.

By using RDA, you can manually add these types of rows:

Row Type	Description
Data rows	Fetch data from Oracle's JD Edwards EnterpriseOne tables. You must define the rows and identify the data using the business view that is attached to the tabular section. Data rows represent groups of data fields that are associated with the columnar amounts. For example, you can have a data row that displays revenue (column) for a range of items. In addition, you can add a row that displays the direct costs (column) for another range of fields.
Calculation rows	Display amounts that are calculated from other rows. For example, you can calculate the gross margin of the revenue and direct costs rows.
Sum rows	Define a special type of calculation. The calculation performs totaling for all numeric columns in a range of rows. The total can include or exclude rows within the sum range that are themselves row calculations.
Underline rows	Enable you to create underlines to separate various rows in the report.
Constant rows	Contain only text. Enables you to describe or label information in the tabular section, such as identification information for a group of rows.
Automatically generated rows	In tabular sections, the system automatically generates rows that define a chart of accounts for a business unit or represent the merging of accounts from several business units. After the generation, you can insert rows that calculate account roll-up totals at various levels of detail. For example, you might want the system to automatically generate rows to create a balance sheet report that is based on the current month and the prior year's account balance.

The system optimizes rows with two or less logical expressions (nodes) that are joined by an AND operator. The optimization accelerates report processing speed. If you include a cover page with the report, you can review which rows the system is optimizing. In the Optimized Row column of the cover page, the system prints a **Y** next to the rows that are optimized and an **N** next to the rows that are not. The cover page also includes the total number of rows that are optimized, Number of Optimized Inclusion Rows, and the total number that are not, Number of Non Optimized Inclusion Rows. You can also find the total number of optimized rows in the UBE log file, Tabular Optimization.

The row optimization feature is enabled by default. As a troubleshooting measure, you can disable row optimization by adding this line to the UBE section of the jde.ini:

```
[UBE] UBETabOpt=0
```

To again enable row optimization, set the variable to **1,** or delete the row from the jde.ini file.

Prerequisites

Before you begin working with tabular rows, ensure that you:

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

Adding Data Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Create, Data.
3. On the Data Row Properties form, enter the name of the data row in the Name field on the General tab.

   The name appears in the Name field on the Row List tab of the Tabular Section properties and on the Data Row Properties form.
4. In the Description field, enter a meaningful description of the data.

   The description appears on the report in the Description column of the Tabular section, on the Row List tab of the Tabular Section properties, and in the Description field of the Data Row Properties form. This Description column can consist of multiple lines to accommodate as much text as required. To indent the text on the report, enter spaces in front of the text.
5. Set other properties that you want on other tabs, and click OK.
6. On the Data Selection form, define the criteria that you want to apply to the data row, and click OK.

   At any time in the future, you can modify the data selection by selecting Define Data Selection from the Row menu.
7. To add rows directly beneath the data row, click the data row (indicated by a black box around the row) and add another row.

Adding Calculation Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Create, Calculation.
3. On the Calculation Row Properties form, enter the name of the calculation row in the Name field on the General tab.

The name appears in the Name field on the Row List tab of the Tabular Section properties and on the Calculation Row Properties form.

4. In the Description field, enter a meaningful description of the calculation.

   The description appears on the report in the Description column of the Tabular section, on the Row List tab of the Tabular Section properties, and in the Description field of the Calculation Row Properties form.
5. Set other properties that you want on other tabs, and click OK.
6. On the Expression Manager form, define the calculation, and click OK.
7. To add rows directly beneath the calculation row, click the calculation row (indicated by a black box around the row) and add another row.

Adding Sum Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Create, Sum Row.
3. On the Sum Row Properties form, enter the name of the sum row in the Name field on the General tab.

   The name appears in the Name field on the Row List tab of the Tabular Section properties and on the Sum Row Properties form.
4. In the Description field, enter a meaningful description of the sum.

   The description appears on the report in the Description column of the Tabular section, on the Row List tab of the Tabular Section properties, and in the Description field of the Sum Row Properties form.
5. In the From Row list, select a row to use as the beginning in a range of rows to be included in the sum.
6. In the To Row list, select a row to use as the ending in a range of rows to be included in the sum.
7. Select the Include Intermediate Calculation option if you want the sum to include calculation rows that reside between the From and To rows.
8. Set other properties that you want on other tabs, and click OK.
9. To add rows directly beneath the sum row, click the sum row (indicated by a black box around the row) and add another row.

Adding Underline Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Create, Underline.
3. On the Underline Row Properties form, you do not need to enter a name in the Name field on the General tab.
4. Set other properties that you want on other tabs, and click OK.

   For example, select the Font/Color tab to modify the color of the line or the Options tab to modify the thickness and spacing of the line.
5. To add rows directly beneath the underline row, click the underline row (indicated by a black box around the row) and add another row.

Adding Constant Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Create, Constant.
3. On the Constant Row Properties form, enter the name of the constant row in the Name field on the General tab.

   The name appears in the Name field on the Row List tab of the Tabular Section properties and on the Constant Row Properties form.
4. In the Description field, enter a meaningful description of the constant.

   The description appears on the report in the Description column of the Tabular section, on the Row List tab of the Tabular Section properties, and in the Description field of the Constant Row Properties form.
5. Set other properties that you want on other tabs, and click OK.
6. To add rows directly beneath the constant row, click the constant row (indicated by a black box around the row) and add another row.

Generating Rows Automatically

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Row menu, select Automatic Row Generation.

   > **Note:** Depending on how the section is designed, some fields might not appear on the form.
3. On the Financial Account Level of Detail Row Generation form, enter the business unit for which you want to generate account information in the Business Unit field.
4. In the From Account field, enter the beginning account number to appear on the report.

   If you leave the From Account field blank, no accounts are generated.
5. In the Thru Account field, enter the ending account number to appear on the report.

   If you leave the Thru Account field blank, no accounts are generated.
6. In the Ledger Type (Optional) field, enter the ledger type to appear on the report.

   If specified, the ledger type is included in the data selection for that row.
7. In the Level of Detail field, enter the level of detail to appear on the report.

   You can generate additional account level of detail rows at level of detail breaks. The default is to generate total rows. The row amounts are based on account ranges that are specified through selection criteria.
8. Select the Totals option to generate totals for each level of detail.
9. Select one of these options under the Add Row Options heading:

   - Replace

     Replaces all previously defined rows with the automatically generated rows. This is the default option.
   - Insert

Inserts automatically generated rows after the row that you clicked before selecting the automatic row generation option.

- Append

  Appends automatically generated rows to the bottom of previously defined rows. Use the Append and Insert options to build hybrid account structures from several different business units.

Overriding Cells and Cell Properties

This section provides an overview of cell properties and overrides, lists the prerequisites, and discusses how to:

- Override cells in data or calculation rows.
- Override data selection in data row cells.
- Override cell properties in underline rows.
- Override cell properties in constant rows.

Understanding Cell Properties and Overrides

Cells contain information that is located at the intersection of a column and a row in tabular sections. Cells are populated based on how you define the row. For example, if you define a row to report an inventory item, one cell might be the item cost of the inventory item.

When you override cell properties, you modify the way the data in the cell appears. When you override a cell, you override the data that is located in the cell.

Cell Property Overrides

You can define cell overrides to override the properties of cells. For example, to emphasize the data in a specific cell, you can override its properties, increase the size of the font, and make the font bold.

Cell Overrides

You can define cell overrides to override the data that appears in the cell. For example, to review the results of possible salary increases for a group of employees, you can override the monthly salary cell by employee to change the annual income calculation. This can help you get an idea of how specific raise amounts or percentages might affect an entire department or business unit.

To override a cell in a tabular section, you must first select Cell Mode from the RDA Cell menu. Cell Mode is different from most menu items in RDA because it is a toggle option. When you click the Cell Mode menu option once, the option appears inset. In cell mode, when you click an individual cell, the cell is outlined by a solid-line rectangle. When you are not in cell mode and you try to click an individual cell, the entire row is outlined by a solid-line rectangle.

When a cell is overridden, it is outlined by a dashed-line rectangle. You can override cells in these types of rows:

- Data rows
- Calculation rows
- Underline rows
- Constant rows

Prerequisites

Before you begin overriding individual cells, ensure that you:

- Create a batch application object.
- Complete the design of the report template using a tabular section with tabular rows.

Overriding Cells in Data or Calculation Rows

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Cell menu, select Cell Mode.
3. On a data or calculation row, click the cell that you want to override.
4. From the Cell menu, select Create Override, Calculation.
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.
7. On the Expression Manager form, define the expression to populate the cell with a new value, and click OK.

Overriding Data Selection in Data Row Cells

Open a tabular row section report in RDA.

1. Click the tabular section.
2. From the Cell menu, select Cell Mode.
3. On a data row, click the cell that you want to override.
4. From the Cell menu, select Create Override, Data.
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.
7. On the Data Selection form, add or modify data selection, and click OK.

Overriding Cell Properties in Underline Rows

Open a tabular section report in RDA.

1. Click the tabular section.
2. From the Cell menu, select Cell Mode.
3. On an underline row, click the cell that you want to override.
4. From the Cell menu, select Create Override, Underline.
5. On the Cell Properties form, modify the name and description on the General tab, if appropriate.
6. Select the Options tab, and modify the thickness on the line.
7. Set other properties on the available tabs as appropriate, and click OK.

Overriding Cell Properties in Constant Rows

Open a tabular section report in RDA.

1. Click the tabular section.
2. From the Cell menu, select Cell Mode.
3. On a constant row, click the cell that you want to override.
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.

9 Modifying the Appearance of Report Sections

Understanding the Appearance of Report Sections

Sections, column headings, column variables, runtime fields, and constants have their own set of properties.

Some properties affect the report section layout. For example, you can modify the format of the report by changing column and row spacing and object alignment.

Using the features of Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA), you can change the appearance of report sections by:

- Changing the description of sections that appear on the section tile.
- Hiding report sections conditionally and unconditionally.
- Aligning fields within and across sections.
- Wrapping the text of fields that are fetched from the database.
- Changing column and row spacing.
- Reprinting data from one page to the succeeding page.
- Inserting page breaks manually.

Working with Section Descriptions

This section provides an overview of section descriptions, lists the prerequisites, and discusses how to change section descriptions.

Understanding Section Descriptions

The RDA workspace uses a grid to present all sections that are included in the report template. The exception might be a section that you set up as invisible. The grid is white and typically displays dots that aid you in positioning fields within a section. To the left of each section is a gray tone tile. This tile displays section descriptions that typically represent either the section type, such as page header, or the name of the attached business view.

A level break section is typically named after the level break field on which it is based. For example, *On Search Type* is the description for a level break header section that breaks on the Search Type field.

If you do not want to view these section tiles, you can hide them in User Options.

When you create a custom section and attach the same business view that you attached to the level one section, both sections have the same description. To tell the sections apart in Event Rules Design, rename one of the sections. You can change the section description on the appropriate section properties form. Adding a prefix to the custom section to indicate that it is the custom section is helpful.

A group section that is created to hold grand totals is untitled because it does not have a business view attached. On the appropriate section properties form, you can modify the section description to read **Grand Totals.**

Prerequisites

Before you begin changing section descriptions, ensure that you:

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

Changing Section Descriptions

Open a report template in RDA.

1. Double-click the section that you want to change.
2. On the appropriate section properties form, enter a new name (or modify the existing name) in the Description field on the General tab, and click OK.

Hiding Report Sections

This section provides an overview of how to hide report sections, lists the prerequisites, and discusses how to hide and display report sections unconditionally.

Understanding How to Hide Report Sections

You can hide a report section if the sole purpose of the section is to provide logic to be used by another section. The information that is contained in an invisible section does not print on the report. Hiding report sections in this manner is referred to as unconditional. The section is *never* meant to appear in the report.

You could open a report template that includes an invisible section and not know that the section exists. So that you can see invisible sections of reports in the RDA grid, select the Show Invisible Sections At Startup option in User Options.

You can also hide a report section based on specific criteria. Hiding report sections in this manner is referred to as conditional because the section appears only when the stated condition is met. You can create event rules that show the section only when a field value reaches a certain amount. Another condition might be when the user requests, through processing options, that the section be included in the report.

See *Setting Up Custom Sections*.

Prerequisites

Before you begin hiding report sections, ensure that you:

- Create a batch application object.

- Complete the design of the report template.

Hiding and Displaying Report Sections Unconditionally

Open a report template in RDA.

1. Double-click the report section that you want to affect.
2. On the appropriate section properties form, select the Advanced tab.
3. Select the Visible option to display the report section, or clear the Visible option to hide the report section, and then click OK.

Aligning Fields and Columns

This section provides an overview of how to align fields and columns, lists the prerequisites, and discusses how to:

- Align fields within sections.
- Align fields and columns across sections.

Understanding How to Align Fields and Columns

When you initially create a report section, fields and columns might not line up properly. RDA provides alignment options to enable you to precisely adjust the appearance of the report output. For example, you create a group section and need to align fields under other fields. Or you create a columnar report with a level break footer that calculates totals. You change the size of the columns in the columnar section, and the total in the level break footer no longer aligns with the column that it is totaling.

When you align fields, you are presented with options based on your selection to align objects within a section or across sections. On the Align Objects form, options are listed under these headings:

Alignment Options	Description
Left to Right	Available for aligning objects within sections and across sections.
Top to Bottom	Available for aligning fields across sections only.

When you align data fields, use these guidelines:

- Clear the focus of the cursor in all report sections before you select the objects for alignment.

  Click an empty portion of each report section to clear the focus of the cursor. You could have the cursor focused on a field in a section, such as the page header, and not realize it. When you align a field without clearing the focus, a field from another section, such as the page header, will be aligned with the selected field.
- Designate one object as an anchor to which other objects are aligned.

  The anchor field is indicated by a black border; the objects to be aligned with it are indicated by a gray border. The anchor field is the *last* field that you select during the alignment process.

- The black border indicates the currently selected field.
- The entire object must be selected for alignment, not just the constant text or variable.
- The anchor can be a disconnected constant or variable object.
- The objects that are selected can be within the same section or across sections.
- The fields are aligned rather than the text within the fields.

  This is especially noteworthy when you center-align fields. Objects are centered based on field length and not on the length of the text within the fields.
- The alignment process cannot be used on tabular rows.

When you align columns from tabular and columnar sections, use these guidelines:

- To align columns with group section objects, select the column in the tabular or columnar section that you want to use as the anchor for the alignment.
- To align columns with columns in another columnar or tabular section, you can select only one column in each tabular or columnar section.

Prerequisites

Before you begin aligning fields and columns, ensure that you:

- Create a batch application object.
- Complete the design of the report template using a columnar section.
- Include a level break footer with a label.

Aligning Fields Within Sections

Open a columnar report template in RDA that includes a level break footer.

1. Click the aggregate label in the level break footer.
2. Hold down the Ctrl key and click the total field.

   The rectangle around the aggregate label lightens in color while the rectangle around the total is dark in color.
3. From the Layout menu, select Align.
4. Select the Current Section option under the Apply to heading.
5. Select one of these options under the Top to Bottom heading:
   - Top Edges
   - Middle
   - Bottom Edges
   - No Changes
6. Click Apply to view the changes without leaving the form.

Aligning Fields and Columns Across Sections

Open a columnar report template in RDA that includes a level break footer.

1. Click the total in the level break footer.
2. Hold down the Ctrl key and, in the columnar section, click the column variable that is associated with the total.
3. From the Layout menu, select Align.
4. Select the All Sections option under the Apply to heading.
5. On the Align Objects form, select one of these options under the Left to Right heading:
    - Left Edges
    - Center
    - Right Edges

      Typically, when aligning numeric values, you want to align by the right edges so that the decimal places line up properly.
    - No Changes.
6. Click Apply to view the changes without leaving the form.
7. When the alignment is complete, click OK.

Using Absolute Position for Text Wrapping

This section provides an overview of absolute position, lists the prerequisites, and discusses how to activate absolute position.

Understanding Absolute Position

The Absolute Position option enables you to wrap text that is fetched from the database, business view columns, and media object text. This option does not affect text that is supplied through event rules or entered into constant fields.

When you select the Absolute Position option, text is truncated in fields where the display length is too short. The system leaves the text in its absolute position. When you clear the Absolute Position option, the text is wrapped to the next line of the field.

> **Note:** You cannot define text wrapping for individual fields; you must apply text wrapping to an entire detail section.

Prerequisites

Before you begin activating absolute position, ensure that you:

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

Activating Absolute Position

Open a report template in RDA.

1. Double-click the section in which you want to wrap text.
2. On the appropriate section properties form, select the Advanced tab.
3. Select the Absolute Position option to inactivate text wrapping or clear the option to activate text wrapping, and then click OK.

   If the Advanced tab does not include an Absolute Position option, you cannot affect text wrapping for the section.

   > **Note:** The Absolute Position option is cleared by default.

Changing Column Spacing

This section provides an overview of column and row spacing, lists the prerequisites, and discusses how to:

- Modify column spacing.
- Modify row spacing in columnar sections.
- Modify row spacing in tabular row sections.

Understanding Column and Row Spacing

You can modify the space between columns and the space between rows in columnar and tabular report sections.

In the Spacing menu option, which is available only for columnar and tabular sections, two tabs are available:

- Column Spacing
- Row Spacing

The Column Spacing options enable you to:

- Center the report under the page header information.
- Modify the spacing among all columns of the section.
- Modify the spacing between individual columns.
- Bring attention to the data in a specific column by adding spaces to set the column apart from the other columns.

On the Column Spacing tab, you can select all columns or indicate specific columns that you want to affect. You enter a value in the Space before selected columns field to indicate the amount of space that you want between the columns. Click Apply to view the new spacing before leaving the form to ensure that the new spacing meets your needs.

You can also change the spacing of rows to improve the appearance of a report. From the Row Spacing tab, you can modify the spacing between:

- The page header and the detail section.
- The rows in the detail section.

Your choices for the row spacing of a tabular or columnar section are:

- Single
- Single + Half
- Double

> **Note:** Only single spacing is supported for CSV files. Only single and double spacing are supported for line printers.

In a tabular row report, you can change the space before a row. This is defined on the Options tab of the Row Properties form.

Prerequisites

Before you begin changing column spacing, ensure that you:

- Create two batch application objects.
- Complete the design of the first report template using a columnar section.
- Complete the design of the second report template using tabular rows.

Modifying Column Spacing

Open a columnar section report in RDA.

1. Click the columnar or tabular section in which you want to modify column spacing.
2. From the Layout menu, select Spacing.
3. On the Column Spacing tab, perform one of these actions to affect the column spacing:
   - Click the Select All Columns button to apply the spacing to all columns in the section.
   - Select specific columns from the list to which you want to apply the spacing.
4. In the Space before selected columns field, enter the number of spaces that you want to appear before the selected columns, and click OK.
   The default value is set at five spaces.

Modifying Row Spacing in Columnar Sections

Open a columnar section report in RDA.

1. Select the columnar section in which you want to modify row spacing.
2. From the Layout menu, select Spacing.
3. On the Columnar Section Spacing form, select the Row Spacing tab.
4. Select a Header to Detail option and a Detail to Detail option.

   > **Note:** Only single spacing is supported for CSV files. Only single and double spacing is supported for line printers.

5. Click Apply, and then click OK.

Modifying Row Spacing in Tabular Row Sections

Open a tabular row section report in RDA.

1. In the tabular section, double-click a row to create a space above it.
2. On the appropriate row properties form, select the Options tab.
3. Enter a value in the Space Before field, and click OK.

Reprinting Information After a Page Break

This section provides an overview of Reprint After Page Break, lists the prerequisites, and discusses how to reprint information on the succeeding page.

Understanding Reprint After Page Break

In some instances, you might need information from one page to appear on the succeeding page. You can define a detail section to reprint the last line of a page as the first line on the succeeding page. Use the Reprint After Page Break option on the Advanced tab of the appropriate section properties form to accomplish this.

Occasionally, the information for a specific level of data continues onto the succeeding page. If you have level break headers defined in the report, the level break header appears on the previous page. You can define the level break header to reprint on the succeeding page using the Reprint After Page Break option on the Advanced tab of the appropriate section properties form.

Prerequisites

Before you begin printing the last line of a page as the first line of the succeeding page, ensure that you:

- Create a batch application object.
- Complete the design of the report template using a group or columnar section.
- Include a level break header in the report.

Reprinting Information on the Succeeding Page

Open a columnar or group section report template that includes a level break header in RDA.

1. Double-click the group, columnar, or level break header section that you want to affect.
2. On the appropriate section properties form, select the Advanced tab.
3. Select the Reprint At Page Break option, and click OK.

Inserting Page Breaks

This section provides an overview of manual page breaks, lists the prerequisites, and discusses how to insert manual page breaks.

Understanding Manual Page Breaks

You can insert manual page breaks (that is, cause the report to stop printing on the current page and start printing on the next page) after report headers, detail sections, and level break footers. You cannot use the page break feature for all three section types in the same report. However, you can use the page break feature in the report header and level break footer simultaneously.

Prerequisites

Before you begin inserting manual page breaks, ensure that you:

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

Inserting Manual Page Breaks

Open a report template in RDA.

1. Double-click the report header, detail section, or level break footer section that you want to affect.
2. On the appropriate section properties form, select the Advanced tab.
3. Select the Page Break After option, and click OK.

10 Modifying the Appearance of Report Objects

Understanding the Appearance of Report Objects

You can modify the properties of objects to change how they look or behave. Here are some considerations regarding properties:

- Some properties are available for all field types.

  For example, you can change the font size of a column heading or column variable.
- Some properties are available only for specific field types.

  For example, you can modify the justification of a column variable, but you cannot justify a column heading.
- Some properties enable you to adapt information to meet company requirements.

  For example, you can change a column heading to reflect specific company or industry jargon.

Modifying Field Lengths and Column Widths

This section provides an overview of field lengths and column widths, lists the prerequisites, and discusses how to:

- Modify the length of fields.
- Modify the width of columns.

Understanding Field Lengths and Column Widths

Changing the length of a field and changing the width of a column do not yield the same results. You can modify the display length of most fields from the appropriate item properties form. In a columnar section, you can change the display length of the column variable but not the column heading. Changing the display length of a field changes the number of characters that the batch engine places in the field. For example, in a report, you include the Address Number field that is defined in data dictionary to accommodate 20 characters. None of the records in the report include an address number that is longer than five characters. In the appropriate item properties form, you can modify the display length of the address number field to five. The address number field displays five characters on the report. However, because the column width is large enough to accommodate 20 characters, you see a large amount of white space following the five characters. To eliminate the white space, you can size the address number column so that it is only as wide as the five-character display length.

Changing the width of a column changes the amount of space that is allotted for displaying data. You can change the size of columns in the Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) workspace. This method of changing the column size does not affect the field length. If you size the column to be smaller than its associated data, the data is truncated. For example, if you change the column width of the alpha name field to display 10 characters, but many of the names in the database are greater than 10 characters, the names are cut off in the report. If the column contains a numeric field, the column contains asterisks rather than truncating the number.

When you modify a column heading and enter text that is too long for the column heading length, the entire column heading appears on the report, oftentimes overlapping other columns. RDA provides two column heading fields for each column so that you can split long column headings into two lines.

When a field is populated with data that is longer than the field length, and the data does not reside in the database, the data is truncated. When a string data type, such as media objects, is fetched from the database, the data text wraps within the column. Text wrapping is controlled by the Absolute Position option on the Advanced tab of the appropriate section properties form.

Prerequisites

Before you begin modifying field lengths and column widths, ensure that you:

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

Modifying the Length of Fields

Open a columnar section report template in RDA.

1. Double-click the column variable that you want to size.
2. On the applicable properties form, select the Display tab.

   If the properties form does not include a Display tab, you cannot change the length of the field.
3. Enter a new length in the Display Length field, and click OK.

   You can click the arrow buttons to increase or decrease the display length.

Modifying the Width of Columns

Open a columnar section report template in RDA.

1. Click the column heading that you want to size.

   > **Note:** The column heading is outlined and includes a small, solid black box on the right edge. This box indicates that you can manually size the object. Otherwise, the column width can not be changed manually.

2. Place the cursor over the small black box until the cursor changes to a horizontal line with an arrow on both ends.

   Ensure that the cursor is not a plus sign with arrows pointing in each direction.
3. Click and drag the black box until the column is the size that you require.

Changing Font Properties

This section provides an overview of font properties, lists the prerequisites, and discusses how to:

- Change font properties of individual fields.
- Change font properties for all fields in a section.
- Change font properties for all fields in a report.
- Create objects with bar code fonts.

Understanding Font Properties

A font is a set of print characters that are of the same size and style. Examples of fonts include Courier New and Arial. Typically, fonts include variations such as bold and italic. You can use multiple fonts in a single report. Some fonts may be converted to PDF, PostScript, or Printer Control Language (PCL) files; line printers are fairly limited in the types of fonts that they can print. Properties such as alignment and spacing are not considered font variations and can be applied separately to text in the report template.

Fonts are classified as either proportional or nonproportional. Proportional fonts include different *pitches* (widths) for different characters. In a proportionally spaced font, the letter *I* is narrower than the letter *B*. Examples of fonts with proportional spacing are Arial and Times New Roman. While proportionally spaced fonts generally create a more visually pleasing document, they can be difficult to align because of the varying widths of characters.

Nonproportional fonts are fonts in which every character has the same width. Most typewriters and line printers use these fonts. Examples of nonproportional fonts are Courier New and MS Gothic.

The PDF generation, PostScript, and PCL conversion can support any font size. For line printers, you should generate the PDF file with nonproportional fonts and a font size of 10.

This table lists base fonts that RDA supports, along with their valid printer types:

Font Faces	Postscript	PCL	Line	PDF
Courier New	Y	N	Y	Y
Courier New - Bold	Y	N	N	Y
Courier New - Italic	N	N	N	N
Courier New - Bold Italic	N	N	N	N
Arial	Y	Y	N	Y
Arial - Bold	Y	N	N	Y
Arial - Italic	N	N	N	N

Font Faces	Postscript	PCL	Line	PDF
Arial - Bold Italic	N	N	N	N
Times New Roman	Y	Y	N	Y
Times New Roman - Bold	Y	N	N	Y
Times New Roman - Italic	N	N	N	N
Times New Roman - Bold Italic	N	N	N	N

You can change font properties at three levels in the report template:

- Field

  When you change the font properties for a field, the change affects only that field.
- Section

  When you change the font properties for a section, the change affects all fields in the section, *except* for those fields that have been changed individually. To apply the changes to all fields in the section without exception, select the Apply settings to all Objects option.
- Report

  When you change the font properties for the report, the change affects all fields in all sections of the report, *except* for the fields in sections that have been changed at the section level and those fields that have been changed individually. To apply the changes to all fields in the report without exception, select the Apply settings to all Objects option.

Font Colors

Although you can select any color from the Font dialog box for viewing reports online, the system supports only 8 colors for PCL, 16 colors for PostScript, and black for line printers. Both PostScript and PDF generation use the *RGB model* for color. PCL uses the *Simple Color RGB* model that provides only 8 colors.

Refer to this color support table, where *Y* is supported and *N* is not supported:

Font Colors	PostScript	PCL	Line	PDF
Black	Y	Y	Y	Y
Blue	Y	Y	N	Y
Cyan	Y	Y	N	Y

Font Colors	PostScript	PCL	Line	PDF
Dark Blue	Y	N	N	Y
Dark Cyan	Y	N	N	Y
Dark Green	Y	N	N	Y
Dark Gray	Y	N	N	Y
Dark Magenta	Y	N	N	Y
Dark Red	Y	N	N	Y
Green	Y	Y	N	Y
Light Gray	Y	N	N	Y
Magenta	Y	Y	N	Y
Olive Green	Y	N	N	Y
Red	Y	Y	N	Y
Yellow	Y	Y	N	Y
White	Y	Y	N	Y

CJK Fonts

Chinese, Japanese, and Korean (CJK) fonts receive special treatment within the universal batch engine (UBE) and for output management. In RDA, you can select any font that is available in the system and assign it to the report, section, or object. However, in a CJK environment, only the fonts that are included in this table are supported when the PDF file is generated:

Language	Font Name
Simplified Chinese	STSong-Light-Acro
Traditional Chinese	Mhei-Medium-Acro
Korean	HYGothic-Medium-Acro
Japanese	HeiseiMin-W3-Acro, MS Gothic (true type font)

Language	Font Name

JD Edwards EnterpriseOne does not support PCL for Japanese, Simplified Chinese, Traditional Chinese, and Korean languages.

Bar Code Fonts

You can use bar code fonts to create bar codes in reports. For example, you can add bar codes to reports that include inventory items.

RDA provides base bar code functionality for reports and other batch job output. RDA supports Code 39 fonts for both PCL and PostScript conversion. The True Type font name is *BC C39 3 to 1 Medium.* Bar code fonts can be viewed in RDA, previewed, and then printed. Bar codes print on both PCL and PostScript printers. Because font vendors do not sell scalable PCL fonts, only fixed-point sizes are supported for PCL. The recommended point size range for PostScript is 8 to 24 points.

To specify a particular bar code, you must enter the encoding for that bar code in the constant properties. The encoding is a series of characters and numbers that are preceded and followed by asterisks (*) to identify the text as a bar code.

After you set up the bar code in the report, you must link the printer font name and the True Type font name to a physical printer in the Oracle's JD Edwards EnterpriseOne Bar Code Support (P986166) application before you print to either a PostScript or PCL printer.

Prerequisites

Before you begin changing font properties, ensure that you:

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

Changing Font Properties of Individual Fields

Open a report template in RDA.

1. Double-click the field in the report template that you want to change.
2. On the properties form, select the Font/Color tab.
3. Change the font properties, and click OK.

Changing Font Properties for All Fields in a Section

Open a report template in RDA.

1. Double-click the report section that you want to change.
2. On the appropriate section properties form, select the Font/Color tab.
3. Change the font properties.
4. To affect all objects in the section without exception, select the Apply settings to all Objects option, and click OK.

To return all of the fields in the section to their default font settings, click the Defaults button.

Changing Font Properties for All Fields in a Report

Open a report template in RDA.

1. From the File menu, select Report Properties.
2. On the Properties form, select the Font/Color tab.
3. Change the font properties.
4. To affect all objects in the report without exception, select the Apply settings to all Objects option, and click OK.

   To return all of the fields in the report to their default font settings, click the Defaults button.

Creating Objects with Bar Code Fonts

Open a report template in RDA.

1. Double-click a constant.
2. On the Constant Properties form, select the Font/Color tab.
3. Complete these fields:

Font Properties	Description
Font	Select the name of the bar code font (for example, BC C39 3 to 1 Medium).
Font Style	Select from Regular, Italic, Bold, and Bold Italic.
Size	Select a font size from the list.

4. Select the Description tab.
5. Enter the correct encoding sequence for the bar code in the Name field, and click OK.
6. View the report in Preview mode to review the bar codes.

   When you submit the batch job, the Job Submission form also displays the bar codes properly.
7. From the Batch Processing Setup menu (GH9013), select Bar Code Support.
8. On the Work With Bar Code Font form, click Add.
9. On the Bar Code Revisions form, complete these fields:

Field	Description
Printer Name	Enter the name of the printer that will be used for printing reports that include bar code fonts.
Printer Definition Language	Select the Post Script or PCL option.

Field	Description
True Type Font Name	Click the True Type Font button, and select an appropriate font from the list.
Printer Font Name	Enter the font name that is given by the font vendor (for example, Code39Three).
Symbol Set ID	If you selected the PCL printer definition language, enter the information as indicated by your PCL printer font vendor.

10. Use a scanner to test the bar code output.

Activating Dynamic Positioning

This section provides an overview of dynamic positioning and discusses how to activate dynamic positioning for a server or client.

Understanding Dynamic Positioning

Dynamic positioning repositions objects in the report to allow for multiple font usage. This makes dynamic positioning most useful when you print reports in multiple languages. For example, you print a report in English, Greek, and Chinese. You can print the English report in Arial, the Greek report in Haettenschweiler, and the Chinese report in SimSun. Because the Haettenschweiler and SimSun fonts are wider than the Arial font, the objects on the report must be repositioned to prevent overlap.

The dynamic positioning feature:

- Enables you to designate a new font when the report is printed on a line printer or in a foreign language.
- Automatically adjusts the width of objects on a report so that they do not overlap when a font substitution occurs.

Dynamic positioning is also useful when you send many reports to a line printer and want them in different fonts so that they are easier to read. When sending reports to a line printer, you must use nonproportional fonts.

Dynamic positioning does not adjust for the *height* of report objects when you substitute fonts. If you use font substitution for a report, ensure that you space the report rows to allow for the different font heights.

To use dynamic positioning:

- Activate dynamic positioning in the jde.ini on the machine where the reports are processed.

  You must enable dynamic positioning on each machine that runs reports that use dynamic positioning. After you activate dynamic positioning on a machine, you should not deactivate it. You can override dynamic positioning for an individual report.
- Define font substitutions for languages and line printers.

  If the report is already formatted in the font that you want, you can disable font substitution for the report. After you disable the font substitution feature and save the report, font substitution remains disabled for the report.

You must reestablish font substitution by clearing the override if you want to use font substitution in the future. You can view a report with font substitutions applied.

Activating Dynamic Positioning for a Server or Client

Using an editor program, access the appropriate jde.ini file, either server or client.

1. Locate the [UBE] section and enter:

   ```
   UBEDynamicPositioning=1
   ```
2. Save and close the jde.ini file.

Defining Font Substitutions

This section provides an overview of font substitutions, lists the prerequisite, and discusses how to:

- View font substitutions by language type.
- Define font substitutions for language and line printers.
- Change font substitutions by language type.
- Override font substitutions for reports.
- Apply font substitutions to report templates.

Understanding Font Substitutions

Font substitution enables you to select a language and the font that needs to be substituted in report templates. Then you indicate a new font to use for the selected language. This substitution does not include font size. When the font substitution is made, dynamic positioning automatically adjusts the position of items on the report to accommodate the width of the new font. It does not adjust the height of items on the report.

Font substitution is most useful for defining fonts in reports that are intended to be:

- Printed in a foreign language.
- Sent to a line printer.

Prerequisite

Before you define font substitutions, ensure that you activate dynamic positioning in the appropriate jde.ini file, either server or client.

Viewing Font Substitutions by Language Type

Access the Batch Processing Setup menu (GH9013).

1. Select Font Substitution by Language Type.

2. On the Work with Font Substitution by Language Type form, enter the language type in the QBE (query by example) line, and click Find.

   The form displays all defined font substitutions for the language type that you entered.

Defining Font Substitutions for Language and Line Printers

Access the Work with Font Substitution by Language Type form.

1. Click Add.
2. On the Font Substitution by Language Revisions form, enter a language in the Language Type field using the appropriate user-defined code.

   If you are defining a font substitution for a line printer, leave this field blank to select the domestic language.
3. In the Original Font Name field, select the font that you want replaced.

   For a line printer, enter ***JDE LINE.**
4. In the New Font Name field, select the new font that you want to use on the report, and click OK.

   For a line printer, you must use a nonproportional font such as Courier New.

Changing Font Substitutions by Language Type

Access the Work with Font Substitution by Language Type form.

1. Double-click the font substitution that you want to change.
2. On the Font Substitution by Language Type Revisions form, modify the New Font Name field, and click OK.

Overriding Font Substitutions for Reports

Open a report template in RDA.

1. From the File menu, select Report Properties.
2. On the Properties form, select the Advanced tab.
3. In the Dynamic Positioning section, click the Apply System Language Font button.

   If the Apply System Language Font button is disabled and the Don't Dynamically Position option is cleared, review the dynamic positioning setting in the jde.ini file.
4. On the Would you like to reposition the report objects based on the System Language Font? message box, click No to ignore the system language font, and click OK.

   When you have saved the report with the override defined, the system does not apply font substitutions to the report.

Applying Font Substitutions to Report Templates

Open a report template in RDA.

1. From the File menu, select Report Properties.

2. On the Properties form, select the Advanced tab.
3. In the Dynamic Positioning section, click the Apply System Language Font button.
4. On the Would you like to reposition the report objects based on the System Language Font? message box, click Yes to apply the System Language Font to the report.

   If a font substitution is defined for the font that is currently used on the report, the system replaces the current font and dynamically repositions objects on the report to accommodate the new font.

Using True Type Fonts

This section provides an overview of True Type fonts and discusses how to assign fonts by report language.

Understanding True Type Fonts

When you submit batch versions, PDF files are generated on the server using True Type fonts. Oracle's JD Edwards EnterpriseOne provides several True Type fonts that are compatible with Arial, Courier, and Times New Roman. In addition to these fonts, you can purchase or download additional True Type fonts.

To download Arial, Courier, and Times New Roman fonts, go to corefonts.sourceforge.net. If you want a PDF to contain Asian and European Long (WGL4) characters, you should purchase or download a font that is similar to Arial Unicode MS.

By default, all PDFs are generated using the Arial font. To override the Arial font, use the Font Override by Language (P98980) program to assign fonts by report language.

Assigning Fonts by Report Language

In the Fast Path field, enter **P98980** to access the Work With Fonts form.

1. Complete the Language field, and click Find.

   The fonts that are assigned to the selected language appear in the detail area.
2. If you want to add a new assignment for a language, click Add.
3. If you want to change an assignment for a language, highlight the appropriate line in the detail area, and click Select.
4. On the Language Font Revisions form, enter the language for which you are adding an assignment in the Language field.

   If you are modifying an assignment, this field is unavailable.
5. Click the Font button next to the Font ID - Report field.
6. On the Font form, select from these font properties, and click OK:

   If you are modifying an assignment, the current font properties are highlighted.

   - Font
   - Font style
   - Size

    - Script

7. To make changes to the fonts that appear on forms and grids, click the Font button next to these fields, make the changes, and then click OK:
    - Font ID - Form.
    - Font ID - Grid.
8. To keep the fonts on forms and grids as they are, without the change, select **System** in the Font list.

Justifying Text in Variables

This section provides an overview of text justification, lists the prerequisites, and discusses how to change text justification for variables.

Understanding Text Justification

Justification refers to how text is presented horizontally within the space allotted for the text. You can affect the justification of text in columns and in most variables. You cannot set text justification for constants or column headings. In RDA, you can select from these justification options:

- Right
- Center
- Left

Justification is relative to the frame of the object. For example, if you center text, the text is centered within the frame of the object. Remember that some data might have spaces either before or after. These spaces affect how the data is justified. For example, the business unit field is a string data type with a character size of 12. If all of the business units in your company use only five characters, the data is held in the database with spaces in front of each business unit value. These spaces are taken into consideration at that the time the system justifies the business unit column.

The batch engine supports left and center alignment for all fonts and languages. Right alignment is fully supported for Japanese 7-, 8- and 9-point size MS Gothic fonts only. For Chinese and Korean fonts, right alignment is not supported.

Bar codes must be left-aligned in RDA.

Prerequisites

Before you begin justifying text, ensure that you:

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

Changing Text Justification for Variables

Open a report template in RDA.

1. Double-click the field for which you want to define justification.

2. On the variable properties form, select the Display tab.
3. Select the appropriate justification, and click OK.

   To return all of the fields to their default justification setting, click the Defaults button.

Changing Numeric Formatting

This section provides an overview of numeric formatting, lists the prerequisites, and discusses how to change the appearance of numeric fields.

Understanding Numeric Formatting

In addition to the font and justification properties, you can modify these additional properties of numeric fields:

- Display decimals

  Enables you to control the number of digits that appear after the decimal of a numeric field.
- Edit code

  Enables you to control:

  - Whether numeric values includes commas.
  - How positive and negative values appear.
  - How currency values appear.
- Decimal scaling

  Enables you to scale numeric fields in tabular sections only.

  See *Defining Decimal Scaling*.

From the Display tab of the variable properties form, you can modify the number of digits that appear in numeric fields and select edit codes.

> **Note:** You can use processing options to control the presentation of positive and negative numbers. Processing options override the settings that are entered in RDA.

Edit Codes

Edit codes are user-defined codes that indicate different scenarios for the presentation of numeric fields. The default edit code for a numeric value is derived from the associated data dictionary item. However, you can override the edit code in the field properties in RDA. The Edit Code table that is presented in this section can help you select the edit code that best meets your business needs.

Not all fields that present a numeric value can be formatted using edit codes. The Company field displays a number, but the field is actually defined as a string data type in data dictionary. Typically, you can select edit codes for fields that are defined as either numeric or currency in the data dictionary.

To select the appropriate edit code for the report, review the Negative Amount Notation column in the Edit Code table that is included in this section and select the appropriate option. The Negative Amount Notation information helps you narrow the possibilities to four codes. For example, if you select a trailing minus sign as your negative amount notation, your search is narrowed to the selection of options J, K, L, or M.

Next, review the Zero Balance column in the Edit Code table that is included in this section and determine whether you want to print zero balances in the report. The Zero Balance information narrows the search to two codes. For example, if you print zero balances, your search is narrowed to the selection of J or L.

Next, review the Commas column in the Edit Code table that is included in this section and determine whether you want to include commas in the numeric values on the report. For example, if you want commas in the numeric value, select J. If not, select L.

This Edit Code table lists the available edit codes and their characteristics:

Edit Code	Commas Y/N	Zero Balance Y/N	Negative Amount Notation
A	Y	Y	Cr
B	Y	N	Cr
C	N	Y	Cr
D	N	N	Cr
J	Y	Y	- Trailing
K	Y	N	- Trailing
L	N	Y	- Trailing
M	N	N	- Trailing
N	Y	Y	- Preceding
O	Y	N	- Preceding
P	N	Y	- Preceding
Q	N	N	- Preceding
R	Y	Y	< >
S	Y	N	< >
T	N	Y	< >
U	N	N	< >

Edit Code	Commas Y/N	Zero Balance Y/N	Negative Amount Notation
1	Y	Y	No sign
2	Y	N	No sign
3	N	Y	No sign
4	N	N	No sign

Prerequisites

Before you begin changing numerical formatting, ensure that you:

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

Changing the Appearance of Numeric Fields

Open a report template in RDA that contains numeric fields.

1. Double-click the numeric field for which you want to change the formatting.
2. On the variable properties form, select the Display tab.
3. Enter the number of decimal places to appear in the Display Decimals field.

   You can click the arrow buttons to increase or decrease the number of decimal places.

   > **Note:** Setting display decimals is ineffective if currency has been enabled for the system.

4. If it is available, select the Edit Code field, use the visual assist to select a formatting style, and then click OK.

   To return all of the fields to their default display settings, click the Defaults button.

Associating Lines and Boxes

This section provides an overview of lines and boxes, lists the prerequisites, and discusses how to add lines and boxes to fields.

Understanding Lines and Boxes

Use boxes to bring attention to specific data; use lines to set off totals and grand totals. You can apply rectangles and other line styles to most fields in a report section. Single and double lines are available to position either above or below fields.

You cannot enclose columns or entire report sections in boxes. However, with some experience, you can use rectangles in group sections to group fields. You must create the group section manually, instead of using the Director, because the rectangles must be added to the section before any fields are included in the section layout.

These line options are available for both constants and variables:

- No Lines

  This option is selected by default for variable fields. You must clear this option to activate the remaining options.
- Single Rectangle
- Single Line Over
- Double Line Over
- Single Line Under

  This option is selected by default for column headings.

Example: Adding Rectangles to Group Sections

Begin by creating a batch application report object.

1. Use the Director initially to create the page header and group section.
2. Select a business view for the group section, and then click Finish on the Navigation Assistant to quit the Director.
3. Click the group section and from the Insert menu, select Constant Field.
4. Position the constant in the group section.
5. Double-click the constant and on the Description tab of the Constant Properties form, enter spaces in the Variable Name field.

   The number of spaces that you enter determines the size of the box. When you begin laying out fields, you might need to adjust the number of spaces.
6. Select the Style tab, and clear the No Lines option.
7. Select the Single Rectangle option, and click OK.
8. Access the Business View Columns Browser, and move fields to the group section, placing them inside the rectangle.
9. Preview the report to review the layout and sizing of the boxes.

   Modify font, font size, and spaces to perfect the layout.

   You can add another rectangle to the right of the current rectangle.

Prerequisites

Before you begin associating lines and boxes, ensure that you:

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

Adding Lines and Boxes to Fields

Open a report template in RDA.

1. Double-click the field for which you want to add lines or boxes.
2. On the appropriate properties form, select the Style tab.
3. If you selected a variable, clear the No Lines option to activate the other options on the form.
4. Select a line style, and click OK.

   To enclose the field in a box, select Single Rectangle.

11 Including Attachments and Comments in Reports

Understanding Attachments and Comments

You can include attachments and comments in reports and batch versions. You add attachments to the report template or version. You add comments to an individual field of the report or batch version.

Both attachments and comments can be used to include detail that is important to you or another report designer. Attachments typically contain information that is relevant to the entire report. Comments include information that is specific to the field to which it is attached.

Attachments

Add attachments to the report template or version using the Attachments tab at the top of the Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) workspace. Use attachments to explain unique features of the report template. For example, you create a report template using two detail sections. You attach a different business view to each detail section and join the sections on common key fields. On the Attachments tab, explain that you used a subsection join to create the report. Indicate the business view that is attached to each section.

The Attachments workspace is split into two panes. The left pane is the icon pane, and the right pane is the viewer pane. These attachment types are available in the icon pane:

- Text
- Image
- OLE

  These files conform to the OLE standard.
- Shortcut
- URL/File

You can use attachments to provide generic help text for reports that need to be company specific or to document changes that you made to a report. You cannot print attachments with a report; you can only view the attachments in RDA. Attachments are saved as media objects. For example, attachments are beneficial in explaining that the format of a report uses a subsection join to present all of the required information.

Comments

Comments are text only and are added to individual fields of the report or batch version. To add comments to a field, right-click the field and select Insert Comment. Use comments to explain unique features of the field. For example, you create two calculation fields in an employee salary report, New Salary and Raise Amount. The new salary is calculated by multiplying the annual salary by a percent. The raise amount is calculated by subtracting the annual salary from the new salary. Insert comments on the New Salary and Raise fields to indicate how these fields were calculated.

Comments can be added to constants, variables, and column headings. When a field includes a comment, a small red triangle appears in the upper-right corner of the field as a visual indicator.

Comments are added to individual fields of the report or batch version. Right-click a field that includes a comment to select:

- Show Comment

  Displays the comment in the RDA workspace. If the comment is already presented, Hide Comment appears on the menu.
- Edit Comment

  Enables you to edit the text of the comment.
- Delete Comment

Adding Attachments and Comments

This section lists prerequisites and discusses how to:

- Add and delete attachments.
- Add, modify, and delete comments.

Prerequisites

Before you begin adding attachments and comments, ensure that you:

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

Adding and Deleting Attachments

Open a report template in RDA.

1. Select the Attachments tab at the top of the RDA workspace.
2. To add text, type the desired text in the viewer pane.

   Use the formatting tools at the top of the viewer pane to format the text.
3. To add an object, right-click in the icon pane.
4. From the pop-up menu, select New, and then select an appropriate option.
5. To remove an object, right-click the object's icon in the icon pane, and select Delete from the pop-up menu.

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.

12 Inserting Header and Footer Sections

Understanding Header and Footer Sections

Header and footer sections are special-purpose sections that contain constant, variable, and runtime fields. They typically contain information regarding the content of the report, the intended audience, or disclaimers.

Because header and footer sections typically provide commentary or system-related information rather than data from tables, they are not associated with business views.

When you create a new report template, you are presented with header and footer section options on the welcome form of the Director. You can also add these sections to report templates from the Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) menu.

When you add header and footer sections to a report template from within RDA, they are added to the bottom of the RDA workspace. However, when you print the report, the sections print in the appropriate order. The properties and format of report headers and footers and of page headers and footers are similar to group sections.

See *Understanding Report Processing*.

Creating Header and Footer Sections

This section provides overviews of header sections and footer sections, lists the prerequisites, and discusses how to:

- Create report headers.
- Create page headers.
- Create page footers.
- Create report footers.

Understanding Header Sections

RDA presents you with two header options:

- Report headers
- Page headers

Report Headers

Report headers appear once at the beginning of the report, before the page header. You can create only one report header in a report template. Use the report header to include information regarding the entire report. Appropriate information for report headers might be the time frame that is covered by the report or the confidentiality of the report.

Page Headers

Page headers appear on every page of the report. You can create only one page header in a report template. The page header is typically populated by the system using runtime fields that capture the company name, report name and description, date, time, and page number. These fields can also be added manually in RDA from the Insert menu.

You can include a page header from the welcome page of the Director. You can also add a page header to an existing report template from the Section menu in RDA. Either way, you can create an empty page header section or allow the system to populate the section automatically.

Understanding Footer Sections

RDA presents you with two footers options:

- Page footers
- Report footers

Page Footers

Page footers appear on every page of the report. You can create only one page footer in a report template. You typically add information to the page footer using constants. A page footer might include a description of the content of the report. Appropriate information for page footers might be the region or business unit to which the data pertains.

Report Footers

Report footers appear once, on their own page, at the end of the report after the last record is printed. You can create only one report footer in a report template. A report footer might include a legal disclaimer or some other text to conclude the report. Appropriate information for report footers might be the name of the company that has exclusive legal rights to the report information.

Prerequisites

Before you begin creating headers and footers, ensure that you:

- Create a batch application object.
- Complete the design of the report template without including header and footer sections.

Creating Report Headers

Open a report template in RDA.

1. From the Section menu, select Create, Headers and Footers, Report Header.

   An icon is added to the Report Tree window, and an empty section is added to the RDA workspace.
2. Click the report header.
3. Select fields from the Insert menu, and drag them to the report header section.

   You can add any type of data field except business view columns.

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

Creating Page Headers

Open a report template in RDA.

1. From the Section menu, select Create, Headers and Footers, Page Header.
2. Select Section Only to create an empty page header to which you can manually add fields.

   An icon is added to the Report Tree window, and an empty section is added to the RDA workspace.

   You can also select Auto Create to allow the system to populate the page header section.
3. Click the page header.
4. Select fields from the Insert menu, and drag them to the page header section.

   You can add any type of data field except business view columns and the Page n of Total runtime field.
5. Double-click the Page Header section to open the Page Header properties form.

   > **Note:** You can add extra white space between the page header and the subsequent report section by placing a blank constant field below the header text.

6. Define section properties as appropriate, and click OK.

Creating Page Footers

Open a report template in RDA.

1. From the Section menu, select Create, then select Headers and Footers, and then select Page Footer.

   An icon is added to the Report Tree window, and an empty section is added to the RDA workspace.
2. Click the page footer.
3. Select fields from the Insert menu, and drag them to the page footer section.

   You can add any type of data field except business view columns.
4. Double-click the Page Footer section to open the Page Footer properties form.
5. Define section properties as appropriate, and click OK.

   > **Note:** You can add extra white space between the previous report section and the page footer by placing a blank constant field above the footer text.

Creating Report Footers

Open a report template in RDA.

1. From the Section menu, select Create, Headers and Footers, Report Footer.

   An icon is added to the Report Tree window, and an empty section is added to the RDA workspace.

2. Click the report footer.
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.

13 Working with Level Break Sections

Understanding Level Break Header and Footer Sections

In a report, a set of records that share the same value for a specific field are said to be in the same *level.* For example, in a report that is sorted by search type, all of the records with the search type of E are in the same level. When the value in the search type field changes, it is referred to in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) as a *level break.*

Level break sections are useful for adding special processing. Two section types, *level break headers* and *level break footers,* make adding processing to level breaks easy.

Level break sections organize records into smaller, more manageable units. You can define any field from the attached business view as a level break field. When the system processes the report, the level break triggers an event, such as the printing of a heading or the calculation of totals. You can also define the level break to perform a page break. For example, you can define the company field as a level break field and a page break field. Every time that the value in the company field changes, a page break occurs and the records for the new company begin on a new page. A field must be defined as a level break *before* it can be defined as a page break.

Level break headers and footers are always associated with a detail section. Level break sections do not have business views attached directly, but rather share a business view with the detail section with which they are associated. Therefore, all fields from the business view that is attached to the detail section are available in the level break sections.

When you create a level break section, you are presented with these options for selecting the level break field:

- All columns
  Displays a list of all available fields from the business view that is attached to the detail section that you selected.
- Only existing sort columns
  Displays the fields that you selected as data sequencing fields when you created the detail section.

If you have not defined a level break field in the data sequence, the field that you select when creating the level break section is defined by the system as a level break field. When you return to the data sequence after creating the level break section, the field is defined as a level break field.

After you create a level break section, you can modify its properties.

You can define level break header and level break footer sections as conditional. Through the use of processing options, you can prompt the user to indicate whether to print the level break section in the report. For example, enter Event Rules Design on the level break section. In the Initialize Section event, call the Hide Section system function to hide the level break section. Then add event rules to show or hide the section, depending on the value that is entered into the processing option. You must have an appropriate processing option attached to the report template.

In RDA, you can:

- Add level break headers.
- Associate descriptions in level break headers.
- Add level break footers.
- Insert descriptions for aggregates in level break footers.

Note:

- *Level Break Processing*.

Creating Level Break Sections

This section provides overviews of level break headers and level break footers, lists the prerequisites, and discusses how to:

- Create level break headers.
- Hide the level break field in the detail section.
- Associate descriptions.
- Create level break footers.
- Insert descriptions into level break footers.
- Create example level break sections.

Understanding Level Break Headers

Level break header sections present descriptive headings that appear *prior* to each level of records. For example, if you define search type as a level break field, a level break header introduces each group of search types. The first level break header instance might indicate that the subsequent records are related to search type C. When the value of the search type field changes, the next level break header instance might indicate that the subsequent records are related to search type E.

Because of their free-form layout, group sections are used for level break headers. You can define more than one level break header in a report template. For example, you can define the search type and business unit fields as level break fields to further organize the data.

When you create a new report template, and you know that you want to include a level break header to organize the data, you typically do not include the level break field in the detail section.

When you insert a level break header into an existing report, you might have the same data field in both the level break header section and the detail section. To avoid having the level break field print in both sections, you can:

- Remove the level break field from the section layout *prior* to creating the level break header.
- Hide the level break field in the detail section.

  You can hide fields from the Advanced tab of the appropriate field properties form. If you hide a constant or column heading, the variable is also hidden.

  To make the field visible again, access the visible option from the Fields tab on the appropriate section properties form. On the Fields tab, fields that are invisible include an icon with the *No* sign. You can toggle fields between visible and invisible from the Fields tab and from the report tree.

When creating a level break header, make sure that you select the Display selected column as part of this section option. This ensures that the field appears in the level break header as the descriptive heading.

In the RDA workspace, the level break header appears within the detail section. The name of the level break header usually begins with *On* and indicates the level break field on which it was created. In the report tree, the level break header appears one branch below the section to which it is attached.

Associated Descriptions

You can add a description to the level break header to make the information more meaningful to the report reader. When you associate a description with a level break field, the system displays the description for the level break field currently in memory. For example, the search type value of E has a description of Employees associated with it. When the level break header is processed, this description prints along with the search type value in the level break header.

Although associated descriptions are used most often in level break headers, you can also associate descriptions in level break footers and detail sections. If a level break footer or detail section includes the search type field, you can associate a description with this field to include the description of the value. In columnar sections, the description becomes the next column following the value field, such as the search type column.

Not every data field has a description associated with it. A business function trigger must be attached to the data item. In data dictionary, you can review the Edit Rule tab of a data item to determine whether it includes a business function trigger. Some examples of data items with business function triggers are:

- Address Number
- Business Unit
- Company
- Search Type

Understanding Level Break Footers

Level break footer sections present information that appears *after* the records that are associated with the level break field. Fields within the level break footer are used for displaying aggregates.

An *aggregate* is an object that holds the results of a calculation using the values in other fields. For example, the calculation could be a sum of values, an average of values, or a count of how many records exist. After adding a level break footer and assigning the totaling conditions to the aggregate object within the level break footer, you might need to change the totaling conditions to meet other reporting requirements. If you are using the level break footer to accumulate totals, the values in those fields are calculated dynamically at runtime.

Due to their free-form layout, group sections are typically used for level break footers. Use the columnar section level break footer option when you want the aggregates but not the detail section to print in the report. The aggregate fields appear in a row beneath the column headings. You can define more that one level break footer in a report template. For example, you can define the search type and business unit fields as level break fields to provide more detailed aggregates.

When defining aggregates, determine the appropriate operator as defined in this table:

Operator	Description
Average of	Reports the average of all the amounts in the column.
Count of	Reports how many entries or records are in the column.

Operator	Description
Maximum of	Reports the maximum amount for a record in the column.
Minimum of	Reports the minimum amount for a record in the column.
Total of	Reports the sum of the values in this column.

The operator that you select determines the available selection of operands. Operands are columns that are included in the report layout.

You can select line types to appear with the aggregates. You can select either or both:

- Single overline
- Double underline

When creating level break footers, the last option is to define whether you want the aggregates to reprint at the page break. This option reprints the aggregates from the previous page onto the succeeding page. This is important when the level of data continues from one page to the next.

In the RDA workspace, the level break footer appears within the detail section. The name of the level break footer usually begins with *On* and indicates the level break field on which it was created. In the report tree, the level break footer appears one branch below the section to which it is attached.

Aggregate Descriptions

When you create a level break footer in a report template, you can add a description to the footer. The description is used as a label to describe the aggregates to the report readers. Depending on the level of detail the aggregates require, you can use:

- Constant fields

  Enables you to create a static label that describes the aggregates.

  For example, a level break footer in an employee listing report calculates salary totals by department. You can add a constant field to the level break footer to include a generic label, such as *Totals by Business Unit.*
- Data dictionary fields

  Enables you to create a more detailed label that describes the aggregates.

  For example, a level break footer in an employee listing report calculates salary totals by department. You can add a data item to the level break footer to include a detailed label, such as *Totals for Business Unit 10.* Data items require you to add an assignment in Event Rules Design to populate the data item variable.

See *Working with Event Rules*.

Level break footers are presented after a level break has occurred in the preceding level. When the system encounters a different value in the level break field, the system calculates and presents the information in the level break footer section. Therefore, you must consider this timing when defining event rule assignments in the level break footer section.

When a level break footer prints, the subsequent level break value is already in memory. If you use a data item as the label in a level break footer, you must use the previous business view column in the assignment. If you use business view columns in the assignment, the system fetches the subsequent level break value that is currently in memory. For example, you use the search type data dictionary field in the level break footer of a report to label the aggregates. The first group of records is for search type E and the second group of records is for search type C. You assign a value to the search type data dictionary field. If you assign the search type business view column to the data item, the search type of C appears in the level break footer for the first group of records because the C search type is currently in memory. If you use the previous business view column, the search type of E appears in the level break footer for the first group of records.

> **Note:** *Level Break Processing*

Prerequisites

Before you begin creating level break sections, ensure that you:

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

Creating Level Break Headers

Open a report template in RDA.

1. Click the detail section to which you want to attach a level break header.
2. From the Section menu, select Create, Level Break Header.
3. On the Level Break form, under the Show heading, select one of these options:
   - All columns
   - Only existing sort columns
4. Designate a business view field as a level break field.
5. Select the Display selected column as part of this section option, and click Finish.

   You can modify the level break header, and even select a different field as the level break field, from the Level Break tab on the level break header section properties.

Hiding the Level Break Field in the Detail Section

Open a report template in RDA.

1. Select the detail section containing the level break field that you want to hide.
2. Double-click the variable or column variable portion of the level break field.
3. On the variable properties form, select the Advanced tab.
4. Clear the Visible option.

Associating Descriptions

Open a report template that includes a level break header in RDA.

1. In the level break header, click the variable portion of the level break field.
2. From the Edit menu, select Associate, Description.
3. Position the Description field to the right of the level break field.
4. Double-click the description field to change the properties.
5. On the Associated Description Properties form, change options as required.

Creating Level Break Footers

Open a report template in RDA.

1. Click the detail section to which you want to attach a level break footer.
2. From the Section menu, select Create, Level Break Footer.
3. On the Level Break Footer form, select the Group Section option, and click OK.
4. On the Level Break form, under the Show heading, select one of these options:
    - All columns
    - Only existing sort columns
5. Select a business view field to use as a level break field, and click Next.
6. On the Aggregations form, select an appropriate Operator and Operand.
7. Select one of these line options under the Item display style heading:
    - Single overline
    - Double underline
8. Select the Reprint section at page break option, if necessary, and click Finish.
9. To add aggregates to the level break footer after it has been created, click the level break footer, and select Add Aggregates from the Section menu.

> **Note:** In the RDA workspace, move the cursor over the aggregate object; the field in the detail section upon which the aggregate's calculations are based changes color.

Inserting Descriptions into Level Break Footers

Open a report template that includes a level break footer in RDA.

1. Click the level break footer section.
2. From the Insert menu, select Constant Field.
3. Position the constant field in the level break footer where you want the description to appear.
4. Double-click the constant field.
5. On the Constant Properties form, change the Variable Name field to describe the aggregates.

Creating Example Level Break Sections

This example illustrates the process of adding a level break header and a level break footer to an existing report. The business unit field is defined as the level break field for both the level break header and the level break footer. Even though an aggregate function is performed on the salary column, that field is not designated as a level break field. Business unit is the level break field for the footer because the report totals all of the salaries for each business unit.

> **Note:** For this example, modify the example report that is described in Creating an Example Columnar Section Report.

See *Creating an Example Columnar Section Report*.

Open the Creating a Columnar Section Report example in RDA.

1. Click the columnar section and select Create, Level Break Header from the Section menu.
2. On the Level Break form, select Business Unit - Home as the level break field.
3. Select Display selected column as part of this section, and click Finish.
4. To include the description of the business unit in the level break header, click the variable of the Business Unit - Home field and select Associate Description from the Edit menu.
5. Position the description field to the right of the Business Unit - Home field.
6. Double-click the Home Business Unit variable in the columnar section and on the Advanced tab, clear the Visible option, and then click OK.

   This hides the business unit field in the columnar section.
7. Click the columnar section and select Create, Level Break Footer from the Section menu.
8. On the Level Break Footer form, select Group Section, and click OK.
9. On the Level Break form, select Business Unit - Home and click Next.
10. To total the annual salaries for each business unit, set the operator to **Total of** and set the operand to Rate - Salary, Annual.
11. Select a single overline, and click Finish.

    > **Note:** The order of the header and footer sections within the columnar section has no bearing on how they appear in the printed report.
12. Select the Preview tab to view the report.

14 Working with Smart Fields

Understanding Smart Fields

Smart fields are data dictionary items with one or more business functions attached. Smart fields enable you to include complex, reusable calculations in detail sections of a report. You can edit the event rules that are generated by smart fields using Event Rules Design. Smart fields are used in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) and are not currently available for use in any other of Oracle's JD Edwards EnterpriseOne tools.

Smart fields are organized in smart field templates. Each smart field template is associated with a particular business view. All smart fields in a template use the same business view columns for data selection. Because smart fields are grouped in smart field templates, you must attach a smart field template to a detail section before smart fields are available for the section layout.

If you attach a smart field template to a section and the section uses a different business view from the one that is required by the smart field, the system provides you with the opportunity to change the business view.

> **Note:** If the business view that is attached to the detail section is not the same as the business view that is required by the smart field, then the smart field that you add to the section might not function correctly.

Using Smart Fields in Reports

This section provides an overview of smart field columns, lists the prerequisites, and discusses how to:

- Select smart field templates in existing reports.
- Insert and delete smart fields.

Understanding Smart Field Columns

You define smart field data dictionary items as glossary group K. Smart fields are designed to retrieve and manipulate specific JD Edwards EnterpriseOne table data. For example, you add the FINRPTAB - Account Balance smart field, which is located in the S09001 - Financial Reporting smart field template, to a report. The smart field creates a column that calculates the account balance as of the specified financial period and fiscal year.

Smart fields mask the creation of event rules that call business functions. Business functions are programs that use data structures to perform functions including, but not limited to:

- Requesting specific data from JD Edwards EnterpriseOne tables.
- Returning data to the established parameters in the data structure.
- Performing some type of calculation or other manipulation on the data.
- Returning the desired information, such as column headings and complex calculations, to the report.

Smart fields are associated with smart field templates. Using smart field templates, you can include complex logic in the report without having to do any programming.

Smart field templates are attached to Report Director templates. When you create a new report template, you can select a Report Director template from the Application Reports drop-down list box on the welcome form of the Director.

When you insert a smart field into an existing report, a director guides you through a linear process to set up the smart field parameters and values. The option to insert smart field columns is unavailable for input if no smart field template is associated with the report section.

When you select a smart field, the event to which the associated logic is written is predefined. The section type in which the smart field can be used is also predefined. If a Director Template does not exist for the section type in which you are working, the smart field option is not available from the Create menu.

> **Note:**
>
> - *Creating Smart Fields*.

Prerequisites

Before you use smart fields in reports, ensure that you:

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

Selecting Smart Field Templates in Existing Reports

Open a report template in RDA.

1. From the View menu, select User Options.
2. On the User Options form, select the Allow Smart Field Template Selection option, and click OK.
3. Double-click the detail section of the report to which you want to add smart fields.
4. On the appropriate section properties form, select the General tab.
5. Select a smart field template from the Smart Field Template field, and click OK.

   Only smart fields that are defined for the section type that you selected appear.

   > **Note:** If the business view that is attached to the section that you selected is *not* the same as the business view that is required by the smart field template, then the smart fields that you add to the section will not function correctly.

Inserting and Deleting Smart Fields

Open a report template with a smart field template attached in RDA.

1. Click the detail section in which you want to insert a smart field.

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.

15 Understanding Advanced Report Enhancements

Advanced Report Enhancements

In addition to the basic functionality of Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA), you can use advanced features to create reports with greater depth of information and more advanced functions.

This table identifies advanced features that are available for use in reports:

Feature	Description
Subsection Joins	Joins two detail sections of a report, each with a different business view attached.
Drill Down	Links the data in the report to the associated interactive application. From Adobe Reader, you can click a report value that is fetched from the database to directly access the detail in the associated application.
Text Attachments	Fetches text that is attached to records in Oracle's JD Edwards EnterpriseOne interactive applications. You can design a report to include text attachments that exist for a record.

16 Joining Detail Sections

Understanding Subsection Joins

Subsection joins provide you with an alternative to creating new business views when existing business views do not meet your business requirement. You can use subsection joins to join two detail sections, each using a different business view, within a report template. This enables you to present additional data in the report using fields that are not included in a single business view.

The two subsection join sections are referred to as the *parent* section (generally accessing a master table) and the *child* section (generally accessing a secondary table). The parent section regulates the processing of the report. After each field in the parent section is processed, all of the corresponding records in the child section are processed.

See *Understanding Report Processing*.

Within a report template in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA), you can:

- Create subsection joins.

  Two methods are available for creating subsection joins.

  - Add the second detail section as a subsection join section.
  - Join two existing detail sections within a report template.
- Modify and sever subsection joins.

  After you create a subsection join, some modification might be required. For example, you can modify section and field properties for subsection join sections. You can also change the fields on which the sections are joined and sever the join between the two sections.

You can create a subsection join between two columnar sections, two group sections, or a columnar and a group section. The subsection join option is not available for tabular sections.

Subsection join reports take longer to process than a report using a joined table business view. For this reason, subsection joins are a great alternative to creating a custom business view when the report is not a frequently processed report.

Creating Subsection Joins

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

- Create subsection join sections.
- Modify and sever subsection joins.
- Join two existing detail sections.

Understanding the Join

The two business views that you use to join two detail sections of a subsection join must share common fields. You must join the two detail sections on at least one common field to establish a link between the two sections.

When selecting business view columns to use in the subsection join, consider which fields are required to ensure that the correct child records are associated with the correct parent records. This might require one field or multiple fields to be selected for the join.

If the records in the two detail sections of the subsection join have a many-to-many relationship, select the Join only on level breaks defined in the parent section option. This option processes the associated child records only after all of the records for the defined level breaks in the parent section have been processed. This action is critical when you are joining tables that have a many-to-many relationship, such as joining a detail file to a transaction file.

For example, you can create a report to review accounts payable and general ledger detail. No business view includes both accounts payable and general ledger information. In a report template, create one detail section using the **A/P Detail Reports (V0411G)** business view, and in another detail section use the **G/L Transaction Detail Report (V0911G)** business view. Because both of these business views include the same document type and document number fields, you can establish the subsection join using these fields.

Prerequisites

Before you begin joining detail sections, ensure that you:

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

Creating Subsection Join Sections

Open a report template in RDA.

1. Click the section to be used as the parent section.
2. From the Section menu, select Create, Sub-Section Join, and select one of these options:
    - Group
    - Columnar
3. On the Business View Selection Option form, select one of these business view options, and click Next:
    - I'd like help finding an appropriate business view.
    - I'll find a business view myself.
4. Select the required business view, and click Next.
5. On the Section Layout form, select the business view columns to include in the child section, and click Next.
6. On the Sub Section Join form, verify that the Join to option under the Parent heading is selected and that the parent section name appears in the field.

    These options are typically already defined by the system.
7. Select a business view column from the Child columns list that is common with a business view column in the Parent columns and variables list.

This is typically a common key field such as the address number field.

8. Select the same business view column from the Parent columns and variables list that you selected from the Child columns list.
9. Select as many fields from both the Child columns list and the Parent columns and variables list to ensure that the correct records are matched in the report.
10. Select the Join only at level breaks defined in the parent section option as necessary, and click Finish.

    The child section resides within the parent section and includes a chain-link icon, indicating that it is joined to the parent section.
11. Define data sequencing for the child section.
12. Modify the fields in the child section, as necessary, to enhance the appearance of the report.

Modifying and Severing Subsection Joins

Open a report template that includes a subsection join in RDA.

1. Double-click the child section, or select Section Properties from the Section menu.
2. Select the Sub Section Join tab, and modify options as necessary.
3. To sever the join, click the No Join option, and click OK.

    Both detail sections are now independent sections.

Joining Two Existing Detail Sections

Open a report template that includes a subsection join in RDA.

1. Click the detail section that you want to perform as the child section.
2. From the Section menu, select Sub-Section Join.
3. On the Define Sub-Section Join form, under the Parent heading, select the Join To option.
4. In the Join To field, select the parent section to which you want to join the child.
5. Select a business view column from the Child columns list that is common with a business view column in the Parent columns and variables list.

    This is typically a common key field such as the address number field.
6. Select the same business view column from the Parent columns and variables list that you selected from the Child columns list.
7. Select as many fields from both the Child columns list and the Parent columns and variables list as you need to ensure that the correct records are matched in the report.
8. Select the Join only at level breaks defined in the parent section option as necessary, and click Finish.

    The child section resides within the parent section and includes a chain-link icon, indicating that it is joined to the parent section.

17 Including Text Attachments in Reports

Understanding Text Attachments

Oracle's JD Edwards EnterpriseOne enables you to attach informational text to records in the database. Users can add text attachments to records from JD Edwards EnterpriseOne interactive applications. An example of a text attachment is a resume that is attached to the address book record of an employee. A popular use of text attachments is a brief description of items that are purchased and entered into the Oracle's JD Edwards EnterpriseOne Accounts Payable application.

Adding Text Attachments to Reports

This section provides an overview of text attachments in reports, lists the prerequisites, and discusses how to add text attachments.

Understanding Text Attachments in Reports

You can design reports to display the text attachments that are attached to records. This enables you to present in a report the same text attachments that are available in the interactive applications. These text attachments are also considered media objects.

JD Edwards EnterpriseOne uses media object data structures to associate records with their text attachments. You can use Oracle's JD Edwards EnterpriseOne Object Management Workbench to view a list of all media object data structures (object type GT). Media object data structures are called in event rules using system functions. After you select a media object system function, you must define these parameters:

- Action

  Select the **<Get Text>** action for a media object text attachment.
- Text

  Select the name of the field that you inserted into the report section to display the text attachment.
- Associated fields

  Map null, literal values or fields from the report template with the data items that are included in the data structure. These fields are required to ensure that the correct text attachment appears with the correct record in the report section.

Prerequisites

Before you add text attachments to reports, ensure that:

- Text attachments exist for the records that are required for the report.

- An existing report includes the records that have text attached, or create a new report that includes the required records.
- You can identify the name of the media object data structure that the interactive application uses to associate text attachments with records.

Adding Text Attachments

Open a report template that fetches the appropriate records in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

1. Select the detail section to which you want to add text attachments.
2. From the Insert menu, select Alpha Variable.
3. Position the variable in the detail section.
4. If the section is columnar, double-click the alpha variable column heading and on the General tab of the properties form, enter a descriptive column name.
5. If the section is columnar, double-click the column variable; if the section is group, double-click the alpha variable field.
6. On the Description tab of the properties form, change the name of the variable by entering a new name in the Variable Name field.

   Change the name to something meaningful that is easy to recognize in Event Rules Design (for example, Attachments).

   This field should relate to the column heading name if the section is columnar.
7. Select the Display tab, modify the Display Length field to a length that accommodates the text attachment, and click OK.

   If the text is longer than the display length, the text wraps.
8. Click the detail section of the report, and select Event Rules from the Edit menu.
9. On the Event Rules Design form, select the **Do Section** event from the pull-down events list box.
10. From the Insert menu, select System Function.

    To include text attachments for specific records, establish If/While criteria. If no text attachment for the record exists or if the record is excluded by the If/While logic, no text prints in the report section.
11. On the System Functions form, expand the Media Objects folder on the Function Selection tab.
12. Select the media object data structure that is attached to the associated interactive application.
13. Select the Parameter Mapping tab.

    The Action and Text parameters typically appear on this screen. The remaining parameters vary depending on the data structure that you select.
14. Define the data structure parameters, and click OK.

    The system displays a media object call on the Event Rules Design form. This call varies, depending on the parameters that you mapped.
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.

18 Working with the Drill Down Feature

Understanding the Drill Down Feature

The Drill Down feature is available in tabular report sections only. This includes application reports that use tabular sections. You cannot create an audit trail for tabular reports containing row or cell specifications or calculation fields.

The Drill Down feature enables you to research beyond the summary information in the tabular report section and into the detail from which the information was derived. For example, in a tabular report that presents unpaid balances for customers, you can use the Drill Down feature to review each unpaid invoice that contributes to the total unpaid balance. Use the Drill Down feature to associate the data in the report with the related detail in Oracle's JD Edwards EnterpriseOne interactive application.

For financial reports, an audit trail is created when the related JD Edwards EnterpriseOne application opens. The audit trail includes detail regarding the data on the report. The audit trail records are static; therefore, the information in the audit trail might differ from the information in the records that you are auditing. For example, if someone posts a transaction after you run the report, the change is immediately reflected in the table, but it is not reflected in the audit trail.

> **Note:** Because the Drill Down feature requires significant system resources, you should activate this feature only when it is required.

Defining the Drill Down Feature

This section provides an overview of how to activate the Drill Down feature, lists the prerequisites, and discusses how to:

- Activate and define the Drill Down feature.
- Review audit trails.
- Purge drill down work files.

Understanding How to Activate the Drill Down Feature

You must activate the Drill Down feature before you can define it. You activate and define the Drill Down feature in tabular sections only. This includes financial reports because they use the tabular section type.

You can also activate and define the Drill Down feature in a batch version. When defining the Drill Down feature in a batch version, you must first override the specifications for the section layout and the event rules.

Activate the Drill Down Feature

You can activate the Drill Down feature during these activities:

- Editing or revising tabular section reports in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

  You can access the Drill Down feature on the Tabular Section Properties form. The name of the tab where the Drill Down feature resides depends on the type of section that you create. For a tabular section, the name of the tab is Financial. For a financial report, the name of the tab is Financial Report.
- Creating a financial report using the Report Design Director.
- Modifying or creating a Report Director template.

Define the Drill Down Feature

Begin defining the Drill Down feature by indicating the JD Edwards EnterpriseOne application where the related detail resides for the data that is presented in the tabular section. As part of the Drill Down feature, you define the:

- JD Edwards EnterpriseOne Interactive application

  This is the name of the application that is associated with the data that is presented in the tabular report section.
- Form

  JD Edwards EnterpriseOne interactive applications can include multiple forms. Select the form to appear when the application is launched.
- Version

  Some JD Edwards EnterpriseOne interactive applications have more than one version. Select the appropriate version to launch. When only one version of the application exists, the system does not present this form.

> **Note:** If the report template has processing options attached, you are prompted to provide the parameters after defining the interactive application.

The next step in defining the Drill Down feature is to map the form interconnect fields in the data structure. The data structure passes field values between the report and the selected interactive application. The data flows in the direction that is indicated by the directional arrows.

Map fields from the business view that is attached to the tabular section to the data items that reside on the selected interactive application form. The fields that you map from the tabular section must be the fields that are required to access the detail in the application. You can verify the fields that are required by launching the interactive application and noting the fields that are required to populate the detail form.

You do not typically need to map all fields that are included in the data structure. After selecting the appropriate fields, you must modify the directional arrows to ensure that the data flows from the tabular section to the associated application.

Review Audit Trails

When reviewing a report online from the web client, a user can click specific data in the report to invoke the Drill Down feature. The data that is clicked must be fetched from the database; the user cannot click data that is calculated using event rules because that data is not entered using the application suite. In Adobe Reader, the cursor changes to a pointing index finger when hovered over a field that can be accessed using the defined interactive application. Using the pointing index finger cursor, the user can click the field, and the system automatically launches the associated JD Edwards EnterpriseOne application.

Using the Drill Down feature establishes an audit trail for financial reports, which you can use to view detail about the data in the report. Tabular reports that use the Drill Down feature and are not financial in nature do not create an audit trail.

Purge the Drill Down Work Files

Each time that you activate the Drill Down feature in a financial report, the system creates a work file. This work file remains in the system until it is manually purged.

The information from the audit trail resides in the Balance Audit Work File-Financial Reporting (F83UI001) table. This work file must be purged periodically to improve processing time. Purge the work file using the Oracle's JD Edwards EnterpriseOne Drill Down Table Purge (P83001) application. You can access the JD Edwards EnterpriseOne Drill Down Table Purge application from both the Microsoft Windows client and the web client. Typically, a system administrator is responsible for purging drill down work files.

Prerequisites

Before you define the Drill Down feature, ensure that you:

- Create a batch application object.
- Complete the design of a financial report template in which you can define the Drill Down feature.

Activating and Defining the Drill Down Feature

Open a financial report template in RDA.

1. Double-click the tabular report section for which you want to activate the Drill Down feature.
2. On the Tabular Section properties form, select the Financial tab.

   If you are creating an application report, the system names the tab the same names as the Report Director template. For example, if you select the Financial Reports template in the Director, the tab is named Financial Reports.
3. Select the Drill Down option, and click the Define button.
4. On the Work With Applications form, click Find.

   You can filter the search by entering search criteria in the QBE (query by example) line.
5. Click an application, and click Select.
6. On the Work With Forms form, select a form and click Select.
7. On the Work With Versions form, perform one of these actions:

   - Click a version, and then click Select.
   - Click Close to avoid selecting a specific version.

     If the report has processing options, the system prompts you to provide the parameters.
8. On the Form Interconnections form, click <NOT Assigned> for the field in the data structure that you need to map.

9. Double-click the object in the Available Objects list that is associated with the field from the application form.

   Fields in the Data Item column of the data structure come from the application that you defined. Mapping the associated field from the report ensures that the value is passed into that field of the application form.
10. Click the directional arrow in the Dir column until it becomes a right arrow.

    The arrow indicates that the data flows from the source (the report section) to the target (the interactive application).

    Ensure that you move the cursor down to the next field in the data structure before you select the next available object.

    > **Note:** Available objects vary for each <NOT Assigned> field, depending on the field type of the data item.
11. When all required fields are mapped, click OK.
12. On the Tabular Section form, click OK to return to the Report Design form.

Reviewing Audit Trails

From the Submit Job-Work With Batch Versions-Available Versions form on the web client, process a batch version of a financial report.

1. From the Form menu, select Submitted Jobs.
2. On the Submit Job-Submitted Job Search form, select the batch version that you processed, and then select View PDF from the Row menu.
3. Within the online report in Adobe Reader, position the cursor over the record that you want to research, and click the value.

   The Adobe Reader displays a message asking if you want to launch the application.
4. On the Adobe Reader form, click Yes to open the application form.

   The form that is associated with this report section displays the record that you selected from the report.
5. Select the record for which you want to display details, and click Select.

   The form that is associated with this report displays the details of the record that you are investigating.
6. When you are finished with the evaluation of the form, click Close.
7. When you are finished with the evaluation of the records, open the report template in RDA, access the tabular section properties, select the Financial tab, and clear the Drill Down option.

Purging Drill Down Work Files

Select Drill Down Table Purge from the Advanced Report Setup menu (GH9141) to access the Purge Financial Reporting Drill Down Work File form.

1. Click Find.

   A list of all existing drill down work files appears in the detail area.
2. Select a file, and click Delete.

19 Working with Database Output

Understanding Database Output

You can use database output to update the database using batch applications. You can attach a database output specification to any report section with an attached business view. The Database Output option is available only for report templates that are defined as update reports.

You can use database output in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) to update, insert, or delete records in Oracle's JD Edwards EnterpriseOne tables. Use the *Insert or Update* special operation to insert a record into a table. If a record with the same primary key already exists in the table, the insert fails and the system updates the existing record instead.

You can use database output to load data into JD Edwards EnterpriseOne tables or into text files. Text files can be specified as either comma delimited or fixed-length text records. The comma-delimited format is useful for transferring data to spreadsheets.

Using Database Output to Update Data

This section provides an overview of database output in JD Edwards EnterpriseOne, lists the prerequisites, and discusses how to:

- Define database output.
- Override environments for database output.

Understanding Database Output in JD Edwards EnterpriseOne

This table describes the tools that you use to perform database maintenance in JD Edwards EnterpriseOne:

Tool	Description
Table Conversion	Use for high-performance SQL table-to-table processing. Enables access to non-JD Edwards EnterpriseOne tables. No reports are created.
Database Output	Use to affect tables in different environments that are defined on the Advanced tab of the Report Properties form. Enables you to send data to text files. Output to the database and reports is processed simultaneously.
Table I/O	Use for transforming input data using event rules. The results can be reviewed before affecting the database. Output to the database and reports is processed simultaneously.

Database Output

Database output is performed for every row of data that is processed in the detail section. All database operations occur on a record-by-record basis using the standard JDE Base middleware Application Programming Interfaces (APIs).

Defining database output involves defining the output and then overriding the environment, if appropriate. If no override is defined, the database output runs against the default data source.

When you design database mappings in batch applications, RDA determines where the system stores the flat files that are produced by the mapping. You can also define a default destination using the UBEDBOutputLocation setting in the UBE section of the jde.ini file. If no location is defined in RDA or in the jde.ini file, or if the location is invalid, the files are stored in the current working directory.

The first step in defining database output is to select an operation for mapping the targets:

- Insert only

  Can be used for tables, but is the only appropriate option for text files.
- Update
- Insert or update
- Delete

Next, you need to select the type of target that you are using:

- JD Edwards EnterpriseOne table.
- Comma-delimited text file.
- Fixed-record length text file.

You can suppress a defined database output:

- Override the database output in the version, and delete the mapping record to process the database output for a specific version of the report.
- Define the criteria using If statements and then using the Suppress Section Write system function to process the database output for specific rows of data.

  The Suppress Section Write system function suppresses not only printed output but also database mappings.

> **Note:**
> - *"Designing Tables in JD Edwards EnterpriseOne", "Working with Table Input, Output" in the JD Edwards EnterpriseOne Tools Table Conversion Guide* .

Prerequisites

Before you begin working with database output, ensure that you:

- Create a batch application object.
- Define the report template as an *Update* report.
- Complete the design of the report template.

Defining Database Output

Open a report template in RDA.

1. Click a detail section of the report, and select Database Output from the Section menu.
2. On the Mapping Targets form, select an operation in the Operation(s) field of the detail area.

   You can double-click the Operation(s) and Type fields to view a list of available options. Double-click the option to select it.
3. In the Type field, select the type of file that you are using as a target.

   You can use the same source type several times using different operations.
4. In the Name field, enter the name of the file that you are using as a target, and click Next.
5. On the Mappings form, for each target column, specify which of the available objects from the report section should be assigned (mapped).

   Double-click a Source Section or a Source Object field to display a list of options.

   You can also enable data dictionary overrides from this form.
6. On the Mappings form, click the Map Same button if you want to map all columns.
7. When you have completed the mapping, click Finish.

Overriding Environments for Database Output

Open a report template in RDA.

1. From the File menu, select Report Properties.
2. On the Properties form, select the Advanced tab.
3. Complete the Target field.

   You can also click the Browse button to select the target environment.
4. Complete the Source field

   You can also click the Browse button to select the source environment.
5. Select the Prompt for overrides at runtime option, and click OK.

20 Working with Event Rules

Understanding Events in JD Edwards EnterpriseOne

As a report is processed, the system pauses at specific points to process attached logic. These points are called *events.* You can use these events to attach logic. In JD Edwards EnterpriseOne, you create logic using event rules. Event rules are logic statements that are processed when an event, such as a page break, is encountered. Events can be attached to the report and to the report components, such as variables, constants, and sections.

The event to which you attach event rules varies depending on the purpose of the event rule and the type of section in which the event occurs. For example, if you are adding event rules to columnar or group sections, you typically use the Do Section event. In tabular sections, you typically attach event rules to the Do Tabular Break, Do Balance Auditor, and Column Inclusion events. When you attach event rules to a variable, you can use the Do Variable event, although typically, you use the Do Section event for more efficient processing. Smart fields that are included in a tabular section are automatically attached to the Column Inclusion event.

Each event is triggered at a different time during the processing of the report. You should become familiar with the process flow of events so that you can determine on which event to attach event rules. If you attach event rules to the wrong event, the appropriate data might not be available to successfully process the event rule logic.

> **Note:**
> - *Understanding Events*.
> - *Understanding Report Processing*.

Creating Event Rules

This section provides an overview of event rules, lists the prerequisites, and discusses how to:

- Create If statements.
- Create simple event rule assignments.
- Create assignments using the Expression Manager.

Understanding Event Rules

You create event rules in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) using the Event Rules Design form. You create event rules by selecting from available components. The event rules are embedded in the report template; they cannot be used anywhere else in the system.

Event rules can be written on any event. However, you must understand the order in which the events are processed to determine the correct event on which to place event rules.

If the event rules affect an individual field, you can place the event rules on one of that field's events or on one of the section events. Typically, event rules are placed on section events for more efficient performance.

This table identifies the types of event rule statements that you can create:

Event Rules Components	Description
If/Else/While Statements	Enable you to define the criteria for which the logic should be processed. The first line of an If/While statement typically begins with the If operator. Subsequent lines begin with either an *And* or an *Or* operator. An else statement is used to link two options. For example: If Search Type is equal to E or Search Type is equal to C <do this> Else <do that>. You can embed multiple If statements in event rules and other statements inside of the if statement.
Assignments and Expressions	Enable you to assign a value to a field. For example: RV Business Unit = PC Business Unit. Also enable you to create expressions to populate fields.
System Functions	Enable you to use special functions that are provided by the system. Only system functions that are relevant to the report component and event are presented for use.
Business Functions	Enable you to use logic that is provided by the system. Business functions are either C business functions or business function event rules that can be called from an event.
Event Rules Variables	Enable you to create variables that can be used inside event rules. These variables are typically used to store values to be used in event rule logic.
Table I/O	Enables you to use special table input and output functions that are provided by the system to fetch and update data.
Report Interconnects	Enable you to call another report from event rules.

Fields that are available for use in event rules are represented by a two-character, alphabetical code. This code indicates the source of the field and determines how the field data is used at runtime. Field types that are available for event rules depend on the fields that are included in the report section.

This table describes the field types that could be available in Event Rules Design:

Prefix Code	Description
BC	Business view columns. Columns that are included in the attached business view. These columns are populated with values from the database when a fetch is performed, and those values are the values that are saved when you add or update.
PC	Previous business view columns. Columns that are included in the attached business view. However, these fields are populated with the previous value for the selected business view field.
PO	Processing option fields. Values that are passed from processing options.
PV	Previous report variables. Fields that are populated with the previous value for the selected report variable.

Prefix Code	Description
RC	Report constants. Column headings in columnar sections and constants in group sections.
RS	Report sections. Values that are not typically used when you are creating event rule assignments.
RV	Report variables. Column variables in columnar sections and variables in group sections.
SL	System values. Values that are fetched by the system and include information such as user IDs, environment names, version names, and report names.
TV	Text variables. Fields that you create from the Section menu prior to creating the event rules. These fields are used in assignments.
VA	Event rule variables. Variables that you define in event rules using data items. These fields are used in assignments for storing values for further processing.

Expression Manager

The Expression Manager enables you to create expressions for event rule assignments. The Expression Manager provides a calculator for creating calculations.

This table describes other functions that are available in the Expression Manager:

Functions	Description
Date	Enables you to calculate data such as days_between, months_between, and date_today.
Time Zone	Enables you to calculate data such as utc_get_day, utc_get_hour, and utc_add_years.
General	Enables you to calculate data such as abs( ), exp( ), and round ( ).
Trig	Enables you to calculate data such as cos( ), log( ), and sin ( ).
Text	Enables you to calculate data such as concat( ), ltrim( ), and substr ( ).

Prerequisites

Before you begin creating event rules, ensure that you:

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

Creating If Statements

Open a report template in RDA.

1. Click the object (section, variable, data item, and so forth) to which you want to attach event rules.

   If you are working with columnar or group sections, click the section.
2. From the Edit menu, select Event Rules.

   You can also right-click the object and select Event Rules.
3. On the Event Rules Design form, select an event from the drop-down menu of the events list.

   For example, select the Do Section event. This event is the most commonly used event in columnar and group sections.
4. From the Insert menu, select If/While.

   You can also click the If/While button on the toolbar.
5. On the Criteria Design form, define the criteria for the event rules.
6. Click OK to save the If statement and return to the Event Rules Design form.

   The If statement appears on the Event Rules Design form.

   You can move lines of the event rules to other locations. Select a line and drag it to the desired position.

> **Note:** Changing the sequence of statements in event rules can result in improper syntax. If you detect syntax errors, you can either disable the event rule and continue, or edit the event rule to eliminate the errors.

Creating Simple Event Rule Assignments

Open a report template in RDA.

1. Click the object (section, variable, data item, and so forth) to which you want to attach event rules.
2. From the Edit menu, select Event Rules.
3. On the Event Rules Design form, select an event from the events drop-down list box.

   The system displays a green plus sign (+) next to events that include attached event rules.
4. Click the existing statement to add a statement to follow it.
5. Select Assignment/Expression from the Insert menu.
6. On the Assignment form, in the To Object list, select the field to which you want to assign a value.

   For example, the to object might be the report variable of a column that you inserted into the report.

   When an object is selected, the selected field appears under the To Object heading. This field is the recipient of the assigned value.
7. Select an object from the From Object/Literal list, and click OK.

   The options that are listed in From Object/Literal depend on the field that is selected as the to object.
8. On the Event Rules Design form, select Save from the File menu.

   You can also click the check mark on the toolbar.

Creating Assignments Using the Expression Manager

Open a report template in RDA.

1. Click the object (section, variable, data item, and so forth) to which you want to attach event rules.
2. From the Edit menu, select Event Rules.
3. On the Event Rules Design form, select an event from the events drop-down list box.
4. Click the existing statement to add a statement to follow it.
5. Select Assignment/Expression from the Insert menu.
6. On the Assignment form, in the To Object list, select the field to which you want to assign a value.
7. Click the □(x) button at the end of the From Object/Literal field.
8. On the Expression Manager form, enter an expression using the fields in the Available Information list, the calculator, and the Advanced Functions.
9. When the expression is complete, click OK.
10. On the Event Rules Design form, select Save from the File menu.

Creating and Using Text Variables

This section provides an overview of text variables, lists the prerequisites, and discusses how to:

- Create text variables.
- Use text variables in assignments.

Understanding Text Variables

You can use text variables to hold a literal string for use in event rules. Text variables are easier to translate into multiple languages and are easier to maintain than hard-coded literal strings.

You create text variables in RDA. The text variables are then available in the Event Rules Design form for use in If statements and assignments for a specific report section. Text variables are embedded in the report template; they cannot be used anywhere else in the system.

Prerequisites

Before you begin creating and using text variables, ensure that you:

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

Creating Text Variables

Open a report template in RDA.

1. Click the section to which you want to add text variables.

2. From the Section menu, select Text Variables.
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.

System Functions	Description
Set User Selection and Set User Sequence.	Enables you to define data selection and data sequencing entered into processing option templates.
Stop Batch Processing and Stop Event Processing.	Enables you to stop processing when unfavorable circumstances are met.
Use Data Sel/Seq From a Section.	Enables you to adopt the data selection and data sequencing from another report section.

Prerequisites

Before you begin calling system functions in event rules, ensure that you:

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

Using System Functions in Event Rules

Open a report template in RDA.

1. Click the object (section, variable, data item, and so forth) to which you want to attach event rules.
2. From the Edit menu, select Event Rules.
3. On the Event Rules Design form, select an event from the events drop-down list box.
4. Click the existing statement to add a statement to follow it.
5. From the Insert menu, select System Function.

   You can also click the System Function button on the toolbar.
6. On the Function Selection tab of the System Functions form, expand each folder to view all available functions.
7. Select an appropriate system function.

   For example, to hide an object, expand the Object folder and select the Hide Object system function.
8. Select the Parameter Mapping tab to define the parameters.

   If you double-click the system function, the Parameter Mapping tab is accessed automatically.
9. On the Parameter Mapping tab, select an appropriate field from the Available Objects list.

   For example, to hide an object, select the object that you want to hide from the Available Objects list.
10. Click the right arrow to move the object to the Value column under Parameters.

    Not all system functions require parameter mappings.
11. Click OK to return to Event Rules Design.
12. On the Event Rules Design form, click the check mark to save and return to RDA.

Creating Event Rule Variables

This section provides an overview of event rule variables, lists the prerequisites, and discusses how to create event rule variables in event rules.

Understanding Event Rule Variables

Event rule variables are objects that you create based on the characteristics of a data dictionary field. However, the variables are not stored in the data dictionary. Rather, each variable exists only within the report where it was created. After you create the variable, it is available for use in event rules.

When you create an event rule variable, you indicate the scope. This table describes the scopes that are available for event rule variables:

Scope	Description
Report	Enables you to use the event rule variable in any event *in any section* of the report. Upon completion of the variable, the system adds a prefix of *rpt* to the variable name.
Section	Enables you to use the event rule variable in multiple events *of the selected section.* Upon completion of the variable, the system adds a prefix of *sec* to the variable name.
Event	Enables you to use the event rule variable *in the event where it was created.* Upon completion of the variable, the system adds a prefix of *evt* to the variable name.

The recommended naming convention for event rule variables is to include:

- The Hungarian Notation at the beginning of the name.
- The data item alias at the end of the name, preceded by an underline.
- No spaces in the name.

For example, an event rule variable that you create using the Address Number data item might be named *sec_mnEmployeeNumber_AN8.* The letters *sec* indicate that the event rule variable is available in all events of the report section. The letters *mn* indicate that the field is a math numeric field type. These prefixes are added to the event rule variable name by the system. *Employee Number* is the name that the creator gave the variable. The characters *AN8* indicate that the variable shares the same characteristics as the Address Number data item.

Note:

- *"Understanding Naming Conventions", "Variable Names" in the JD Edwards EnterpriseOne Tools Development Standards for Business Function Programming Guide* .

Prerequisites

Before you begin creating event rules variables, ensure that you:

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

Creating Event Rule Variables in Event Rules

Open a report template in RDA.

1. Click the section to which you want to attach event rule 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.

   This step is required *only* if the event rule variable is used for a specific event. If the variable is required for the entire section or report, you do not need to select an event prior to creating the variable.
3. From the Insert menu, select Variables.

   You can also click the Variables button on the toolbar.
4. On the Event Rules Variables form, enter the alias of the appropriate data item in the DD Item field, and click Add.

   Select a data item that possesses the characteristics that are required by the variable.

   If you do not know which data item you need, you can leave the DD Item field empty, and click Add. You can then use the Variable Options Selection form to search for an appropriate data item.
5. On the Variable Options Selection form, select an appropriate Scope option.
6. In the Name field, enter a meaningful name using the recommended naming conventions, and click Finish.

   The Event Rules Variables form displays the event rule variable that you just created.
7. To modify the scope of the event rule variable, click the variable that you want to edit, and then click the Edit button.
8. On the Variable Options Selection form, make any modifications that are required, and click Finish.
9. Click OK to return to Event Rules Design.
10. On the Event Rules Design form, click the check mark to save and return to RDA.

Using the Do Section Event Versus the Column Inclusion Event

This section provides an overview of the Do Section and Column Inclusion events, lists the prerequisites, and discusses how to use the Column Inclusion event.

Understanding the Do Section and Column Inclusion Events

The most common event for a group or columnar section is the Do Section event. When a group or columnar section is processing data, the Do Section event occurs after each record is fetched.

The most common event for a tabular section is the Column Inclusion event. Because of the automatic totaling on a level break, tabular sections are processed differently. In a tabular section, the Do Section event is processed only at each level break. Therefore, the Column Inclusion even is used in a tabular section to process the data after each record is fetched, rather than waiting until the level break.

Regardless of the section type, do not use the Column Inclusion event when you are performing a calculation between columns (such as when calculating variances) or between variables within a column. For these situations, use the Do Section event.

When using the Column Inclusion event, you attach event rules to individual columns in the section.

Prerequisites

Before you begin using the Column Inclusion event, ensure that you:

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

Using the Column Inclusion Event

Open a tabular section report template in RDA.

1. In the tabular section, click the column variable to which you want to attach event rules.
2. From the Edit menu, select Event Rules.
3. On the Event Rules Design form, select Column Inclusion from the events drop-down list box.
4. Create event rules using the appropriate event rule components.

   For example, to create an if/while statement, select If/While from the Insert menu.
5. Click the check mark to save and return to RDA.

Creating Custom Sections

This section provides an overview of custom sections, lists the prerequisites, and discusses how to set up custom sections.

Understanding Custom Sections

Custom sections enable you to control, through event rules, the information that prints on a report. You can include business view fields, variables, and data dictionary fields in custom sections. The number of custom sections that you can add to a report is limited only by your system's performance. Custom sections are allocated and processed like any other section.

This table describes uses for custom sections:

Custom Section Usage	Description
Force page breaks.	Create a custom section with no objects and then activate the Page Break After Print option in that section.
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 *only* 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.

Note:

- *"Reviewing Batch Version Processing", "Accessing BrowsER for Batch Versions" in the JD Edwards EnterpriseOne Tools Batch Versions Guide* .
- *"Using BrowsER" in the JD Edwards EnterpriseOne Tools Event Rules Guide* .

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.

ER Compare

JD Edwards ER Compare tool provides a side-by-side, visual comparison of two versions of a report's event rules. For example, ER Compare lets you compare a modified report to the original version of that report on the server or in an ESU.

See *"JD Edwards EnterpriseOne ER Compare" in the JD Edwards EnterpriseOne Tools Software Updates Guide* .

21 Understanding Events

Events

You use event rules in Oracle's JD Edwards EnterpriseOne to create logic statements without the difficult syntax required by most programming languages. Event rules process when an event, such as a page break, is encountered.

Events execute in a specific order during the processing of the report. Events are attached to objects, such as fields, sections, and reports. Where the cursor is positioned in the report when you access Event Rules Design determines the set of events that is presented.

Processing Option Logic

When you create the logic that the system uses to manage processing option values, you must consider the purpose of the processing option attached to the report template. The purpose of the processing option helps you to determine on which section and which event to place the logic.

Some processing options must be read before fetching data from the database because they affect which records are fetched. Other processing options might determine whether a level break section appears in the report. You should understand the process in which the system processes sections and events so that the logic provides the intended result.

If the processing option template determines the data to be fetched from the database, you typically place the associated event rules in the Initialize Section event of the affected detail section. For example, if the processing option is prompting for which fiscal year to present, this value must be read before the system fetches any values for that section.

If the processing option determines whether to display a level break section in the report, the processing option needs to be processed before the level break section is processed. The event rules for this processing option are typically placed on the Initialize Section event of the level break section affected.

Event Levels

The section type that you selected and the object on which the cursor is positioned in the report template when you enter Event Rules Design determine the events available. The selection of events is determined by these three levels:

- Reports
- Sections
- Objects

The tables in this section list the events that are available at each of the three levels.

Report Level Events

This table describes the events that are available at the report level; *Y* indicates that an event is supported:

Event	Description
Do Initialize Printer	Y
Initialize Report	future functionality
End Report	future functionality

Section Level Events

This table lists the events that are available at the section level for each section type; *Y* indicates that an event is supported while *N* indicates that an event is not supported:

Event	Report Header	Page Header	Columnar	Group	Tabular	Child/ Custom (CG)	Level Break Header	Level Break Footer	Page Footer	Report Footer
Advance Section	N	N	Y	Y	Y	Y	Y	N	N	N
After Last Object Printed	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Before Level Break	N	N	Y	Y	Y	Y	N	N	N	N
Do Balance Auditor	N	N	N	N	Y	N	N	N	N	N
Do Section	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Do Tabular Break	N	N	N	N	Y	N	N	N	N	N
End Break Section	N	N	Y	Y	Y	Y	N	N	N	N

Event	Report Header	Page Header	Columnar	Group	Tabular	Child/ Custom (CG)	Level Break Header	Level Break Footer	Page Footer	Report Footer
End Lvl Brk Footer Section	N	N	N	N	N	N	N	Y	N	N
End Lvl Brk Header Section	N	N	N	N	N	N	Y	N	N	N
End Page Header	N	Y	N	N	N	N	N	N	N	N
End Report Header	Y	N	N	N	N	N	N	N	N	N
End Report Footer	N	N	N	N	N	N	N	N	N	Y
End Section	N	N	Y	Y	Y	Y	N	N	N	N
Init Break Section	N	N	Y	Y	Y	Y	N	N	N	N
Init Lvl Brk Footer Section	N	N	N	N	N	N	N	Y	N	N
Init Lvl Brk Header Section	N	N	Y	Y	N	Y	Even though event rules can be edited in Report Design Aid, this event is not processed by the system.	N	N	N
Initialize Page Header	N	Y	N	N	N	N	N	N	N	N
Initialize Page Footer	N	N	N	N	N	N	N	N	Y	N
Initialize Report Header	Y	N	N	N	N	N	N	N	N	N

Event	Report Header	Page Header	Columnar	Group	Tabular	Child/ Custom (CG)	Level Break Header	Level Break Footer	Page Footer	Report Footer
Initialize Report Footer	N	N	N	N	N	N	N	N	N	Y
Initialize Section	N	N	Y	Y	Y	Y	Y	Y	N	N
Refresh Section	N	N	N	N	N	Y	N	N	N	N
Suspend Section	N	N	Y	Y	Y	Y	Y	Y	N	N

Object Level Events for Variables

This table lists the events that are available at the object level for variables in each section type; *Y* indicates that an event is supported:

Event	Report Header	Page Header	Columna	Group	Tabular	Level Break Header	Level Break Footer	Page Footer	Report Footer
Cell Inclusion	N	N	N	N	Y	N	N	N	N
Column Inclusion	N	N	N	N	Y	N	N	N	N
Do Variable	Y	Y	Y	Y	Y	Y	Y	Y	Y
End Column	N	N	Y	Y	Y	N	N	N	N
End Variable	Y	Y	Y	Y	Y	Y	Y	Y	Y
Initialize Column	N	N	Y	Y	Y	N	N	N	N
Initialize Variable	Y	Y	Y	Y	Y	Y	Y	Y	Y

Object Level Events for Constants

This table lists the events that are available at the object level for constants in each section type; *Y* indicates that an event is supported:

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

Note:

- *"Using Event Rules Design" in the JD Edwards EnterpriseOne Tools Event Rules Guide* .

22 Understanding Report Processing

Batch Processing

Reports run using a batch process; they processes automatically without user interaction. When the system pauses, the engine executes the logic attached to that event within the batch process. After a batch process is launched, the flow of the attached logic proceeds on a fixed path, which is based on the data being processed. If you need to change the flow of the logic, you must modify the batch process using Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA). Examples of batch processes include reports, subsystem jobs, database output, and table conversions.

Subsystem jobs are batch processes that constantly run in the background and off-load processor resources. Subsystem jobs can also be used to move activities through a process, such as an escalation process in Oracle's JD Edwards EnterpriseOne Workflow, which moves unanswered messages from one user to another after a defined period of time.

See *Working with Subsystem Jobs*.

Use the database output function within RDA to update or insert records within tables.

See *Working with Database Output*.

You use table conversions to:

- Transfer data from one table to one or more tables.
- Change the data or schema of an Oracle's JD Edwards EnterpriseOne table.
- Transfer data from a single business view to one or more tables.

Batch Processes

You use RDA to create reports and batch processes. You can associate individual reports and batch processes with applicable menus to allow users access from the Oracle's JD Edwards EnterpriseOne web client.

You use batch processes to update tables or to print reports indicating the results of the batch process. Reports are just one kind of batch process that generates output. Many batch processes do not generate output.

JD Edwards EnterpriseOne reports contain all of the specifications for a report: section layout, business views, event rules, data selection, data sequencing, and database output.

Each report includes one or more sections; sections are self-contained elements that are the building blocks of reports. You can join sections to one another or use them as standalone groups of information. You can also use sections for special purposes, such as headers and footers.

Section Processing

JD Edwards EnterpriseOne report sections include headers, footers, and detail sections. Detail sections can be columnar, group, or tabular. Detail report sections can be considered mini reports or batch processes, with each section using its own business view. Because other sections in the report can contain different business views, you must add data sequencing and data selection to each section using RDA.

Some report sections are dependent on other sections of the report. Dependent sections provide supporting information and do not necessarily make sense on their own. Other report sections are independent and address the business purpose of the report.

Independent sections are also referred to as level-one sections, and they include these types:

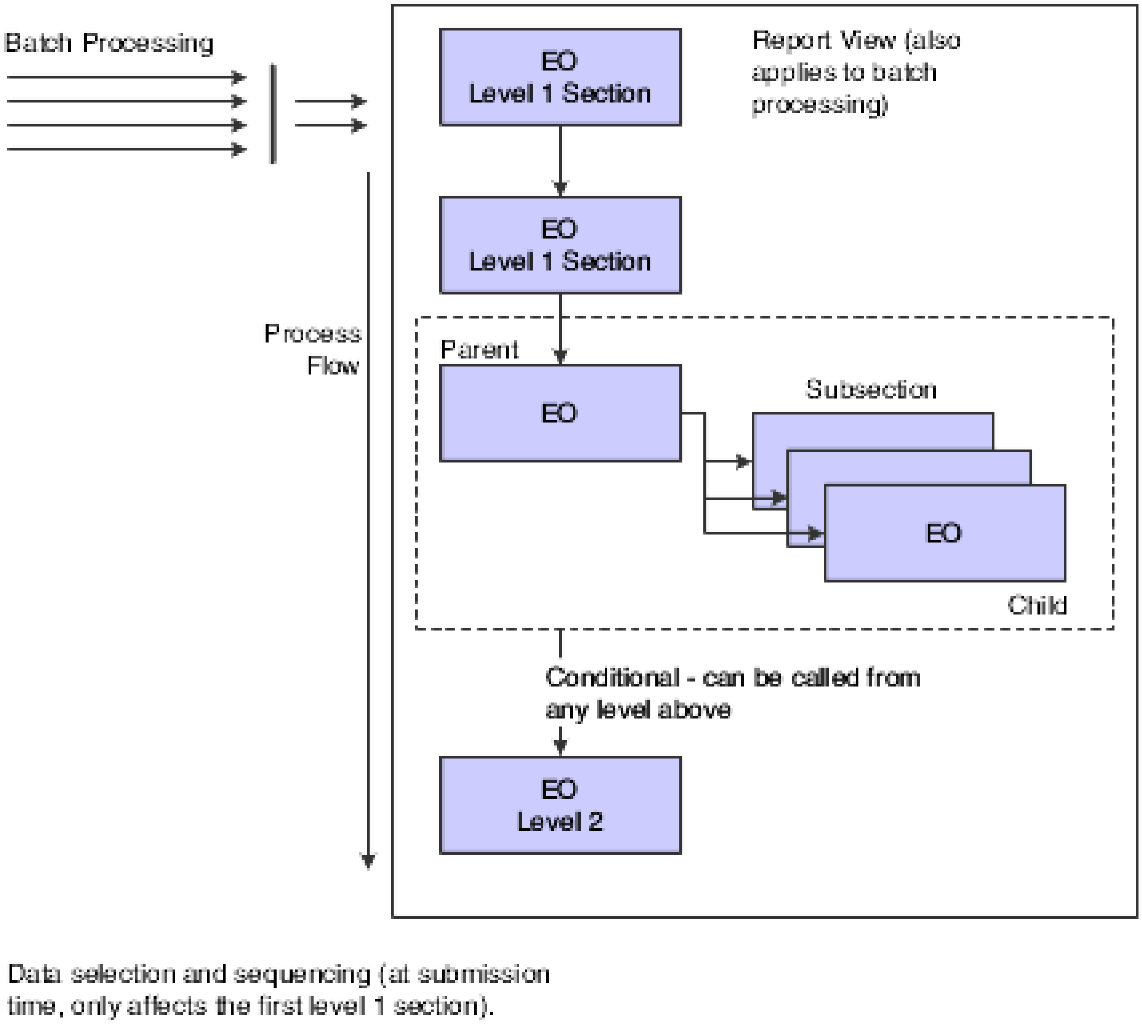
- Group
- Columnar
- Tabular

Level-one sections can also be parent sections.

Dependent sections are called from a level-one section. Dependent sections are also referred to as level-two sections, and they can contain:

- Level break headers.
- Level break footers.
- Total sections (used in non-tabular reports).
- Subsections.
- Customized content.

The order in which the sections appear in the RDA workspace is not necessarily the order in which the system processes them. This diagram illustrates how the batch engine processes sections within a report:


Batch Processing
Report View (also applies to batch processing)
EO
Level 1 Section
EO
Level 1 Section
Process Flow
Parent
EO
Subsection
EO
Child
Conditional - can be called from any level above
EO
Level 2
Data selection and sequencing (at submission time, only affects the first level 1 section).

Processing Sequence

The sequence in which the level-one sections of the report appear in the RDA workspace determines the process flow of the level-one sections. For example, if a report has multiple level-one sections and each has its own business view, then the first level-one section executes before the next level-one section executes. Any level-two sections that exist between the level-one sections are processed when and if they are called from a level-one section. If the report includes a subsection join following the level-one sections, which creates a parent/child relationship, then the parent/child sections are processed independently of the first level-two section.

If level-one sections are moved within the report template, the execution sequence is affected. In report processing, the system processes all level-one sections in the order in which they appear. Level-two sections, however, are processed as dependents of level-one sections. Moving conditional sections, subsections, level break sections, page headers, page footers, and report headers does not affect the sequence of execution. Report execution takes place from top-to-bottom for sections; and from top-to-bottom and left-to-right for objects within a section.

The system reads and processes all records in a section from beginning to end based on the defined data selection. If you do not specify data selection, then the system reads all of the records in the table until it reaches the end of the file.

> **Note:** Data sequencing and data selection defined at runtime affects only the first level-one section of the report or batch process. Therefore, when you design a report or batch process, consider how users will be affected since they cannot easily change data selection and sequencing for additional level-one sections. A level-one section is executed at least once whether or not it has a business view attached to it. Consider this fact in your decision to execute special event rule logic using the Do Section event of that section.

Subsection Joins

The parent/child relationship shown in the previous graphic illustrates the flow of joined sections. Parent/child sections can be called at any point within the process flow. Depending on how the parent and child sections are joined (one-to-one, one-to-many, or many-to-many), records are fetched first from the parent business view and then from the child business view for each corresponding parent record. When all parent records have processed, the system continues to the next section. If any database updates are performed in one of the first level-one sections, that change is reflected in the records fetched for the parent/child sections.

If the report contains a subsection join following a level-one section, then the parent/child section is processed independently of the first section.

Conditional Sections

Shown as the last section in the preceding graphic, the conditional section can be called from any of the previous level-one sections using the Do Custom Section system function. Conditional sections are called from a level-one section dependent on the criteria that you defined. For this reason, conditional sections are considered level-two sections. Memory allocation for conditional sections occurs at the beginning of a section. Therefore, you need to place conditional logic in the Initialize Section or End Section event to hide and show objects because memory is allocated and freed *only once* instead of each time a section is called.

Custom sections appear as level-one sections in the RDA workspace, but they are not processed unless explicitly called by the Do Custom Section system function.

Tabular Sections

Tabular sections output data to the report only when the system encounters a level break. The system does not initialize the Do Section event for each record in tabular sections; instead, it summarizes the records to the lowest level break level. The output to tabular sections is similar to the output to level break header sections.

Section Event Processing

When a report is processed, several events are encountered. The processing of attached logic might be dependent on processes that occur before and after a particular event.

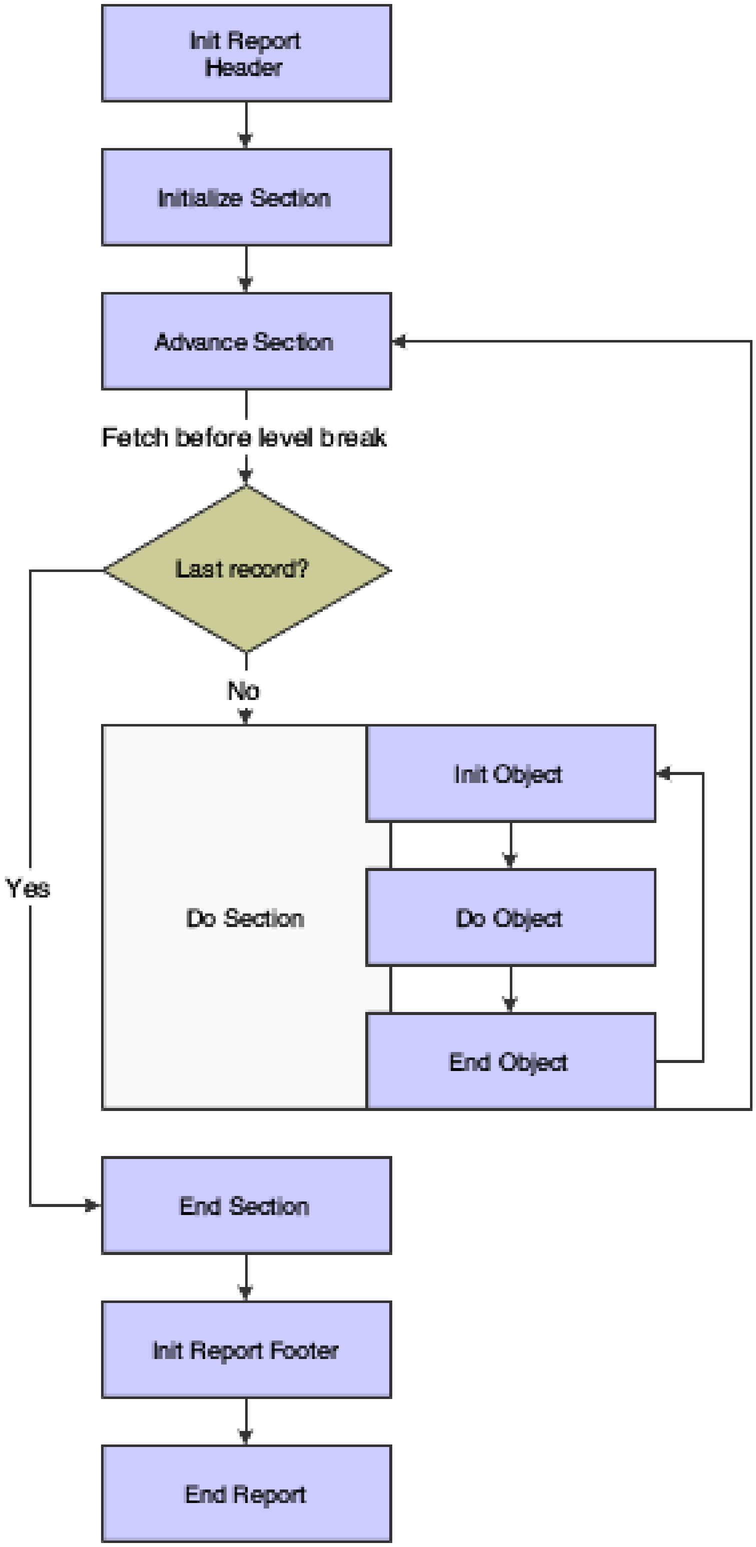
When you submit a report for processing, the system processes the report header before the level-one section, and then the level-one section is initialized by the Initialize Section event. After the Initialize Section event, the system processes the Advance Section, Page Footer, Page Header, and so on. When the system finishes processing the first level-one section and all dependent sections associated with it, the system repeats the process for the next level-one section.

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:

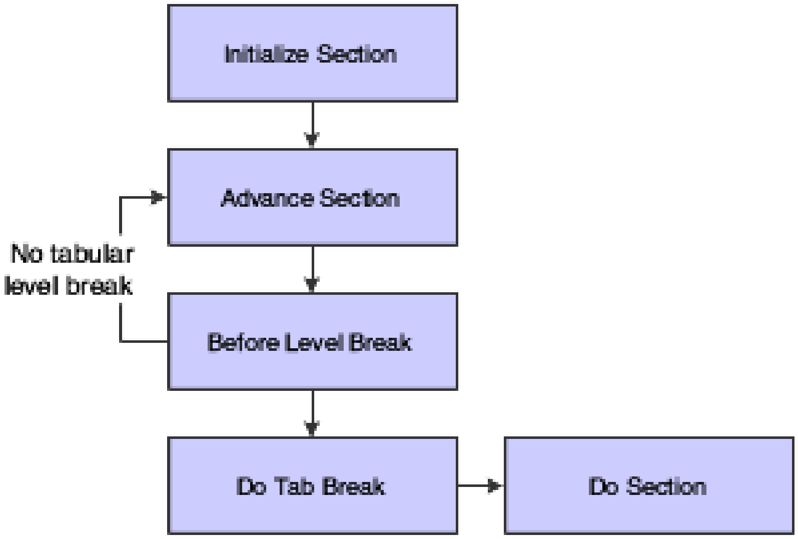

Init Report Header
Initialize Section
Advance Section
Fetch before level break
Last record?
No
Yes
Do Section
Init Object
Do Object
End Object
End Section
Init Report Footer
End Report

Tabular Section Event Flow

Tabular sections are comprised of columns, rows, and cells. You can use calculations to define the information contained in each column, row, or cell. You can use column inclusion event rules to define the information to be contained in each column.

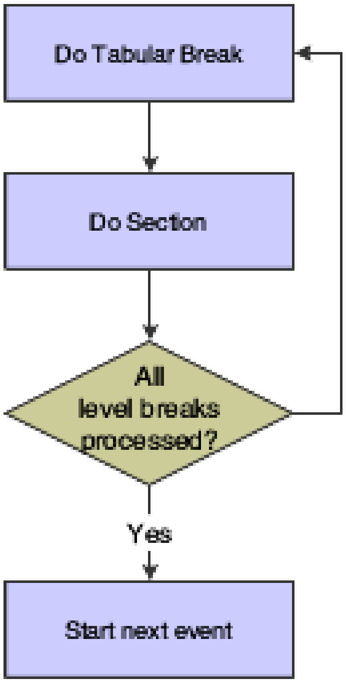
You can include a set of criteria, a business function, or a named event rule in column inclusion event rules. One advantage of tabular sections is that you do not have to define additional sections for level break logic or processing, as you do with group and columnar sections. However, totaling and Row Description columns in tabular sections are dependent on the level breaks that you define. Pay close attention when defining data sequencing and level breaks in tabular sections.

This diagram illustrates the typical event flow for tabular sections:



Tabular sections summarize information and write to the output only when a level break is encountered. Level breaks are dependent on data sequencing and can be modified to make records as detailed or as general as the business need requires. The more fields that you define as level break fields, the more detailed the report. For example, you create a tabular section report to present open purchase orders. You sequence the data on Address Number, Document Number, and 2nd Item Number. You define all three fields as level break fields. The report displays the amount open for each 2nd Item Number for each document for each supplier. If you modify the data sequencing to include only Address Number and Document Number, and define only Document Number as a level break field, the report rolls up all the information and displays the total amount open for each document number. The result is a summary of the data included in the first report.

This diagram illustrates a tabular section level:



Additional features in tabular sections that affect processing include:

- Drill Down

  The Drill Down feature provides a view to the associated interactive application from the viewable output of a report. The interactive application displays the detail for the balances on the viewable output of the report.
- Account Level of Detail
- Automatic Totaling

  The Automatic Totaling feature totals numeric values, regardless of the type, unless otherwise defined in the data dictionary. An example of a data item that does not total is Address Number. However, the system displays the description of some numeric values in the Row Description column, and might try to total these fields. You can suppress totals for a column from the column properties form.

If you have a report that uses multiple sections or a combination of group and columnar sections, consider using a tabular section. Tabular sections improve system performance because the system calls only one section instead of multiple sections.

Logic Attachment to Joined Sections

If you need to include logic on joined sections that should be processed every time the section is processed, attach the logic to the Refresh Section event. When joined sections are processed, the system initializes the Initialize Section event the first time the parent/child section is processed. The system then initializes the Refresh Section event for all subsequent times the parent/child section is processed. Therefore, if you attach logic to the Initialize Section event, the logic is processed only once. If you attach logic to the Refresh Section event, it is processed each time the section is processed.

Data Selection and Sequencing

If a report or batch process contains multiple sections and takes a long time to process, check the data selection for each section. The system processes all the records in a table unless you specify otherwise.

If you want a level-one section to adopt the data selection and data sequencing from another section, you can use the Use Data Sel/SeqFromASection system function. This system function uses the data selection and data sequencing that is defined in another section.

For example, in the Oracle's JD Edwards EnterpriseOne Print Pick Slips (R42520) application, you can select from hundreds of available columns in the Sales Order Detail File (F4211) table to use in the data selection. In the first section, which is hidden and sends data to the F4211 table, the report processes commitments first, which could add rows to the F4211 table. The next section in the report displays the modified and updated F4211 table showing the committed records. Possible solutions include:

- Using a temporary file.
- Changing the data selection on multiple sections.
- Using the Use Data Sel/SeqFromASection system function to adopt the data selection and data sequencing from the previous level-one section.

The Use Data Sel/SeqFromASection system function enables a detail section to adopt the data sequencing or data selection specifications from:

- Another section in the report.
- A different report.

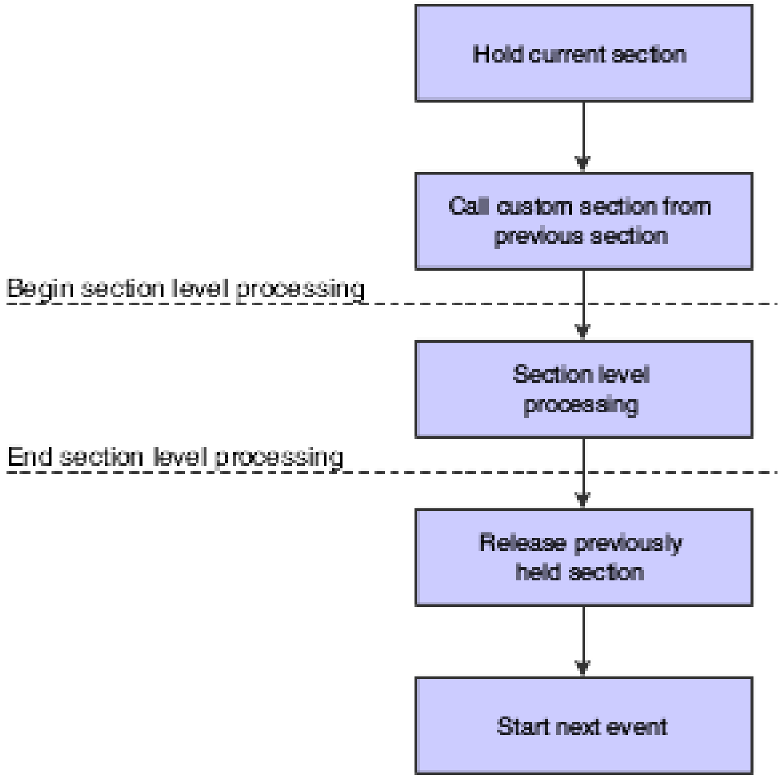
The target section can adopt the data selection criteria, the data sequencing information, or both from the source section. The selection or sequencing information from the source section replaces the information contained in the specifications of the target section.

To use the Use Data Sel/SeqFromASection system function, access Event Rules Design on the target section and access the Initialize Section event. The Initialize Section event is the only event that can invoke this system function. Expand the General folder to locate and select the Use Data Sel/SeqFromASection system function. Define the parameters to indicate the report, version, and section of the source data selection and data sequencing.

Custom Sections

Custom sections enable you to use event rules to control the information that prints on a report. Using custom sections, you can force a page break by creating a section with no objects and then activating Page Break After Print in the Section Properties. You can also use custom sections to print variable text and to present the same information but in different formats. An example would be a report that exists in two different modules but, depending on the user, calls a different section to present information specific to a particular module.

This diagram illustrates the process flow for custom sections called from event rules:


Hold current section
Call custom section from previous section
Begin section level processing
Section level processing
End section level processing
Release previously held section
Start next event

When you run a report that includes a custom section, the batch engine calls and processes each section until it encounters the system function call to the custom section. The batch engine then processes the custom section. When the batch engine finishes processing the custom section, it continues processing the previous section.

You can call custom sections from any event except:

- Initialize Section
- Initialize Column
- Report Level Events
  - Do Initialize Printer
  - Initialize Report
  - End Report

If you try to call a custom section from one of the exception events, the report writes a warning to the log, ignores the call to the custom section, and continues with the main section.

The custom section processes in the same manner as the type of section that it is (columnar, group, or tabular).

Level Break Processing

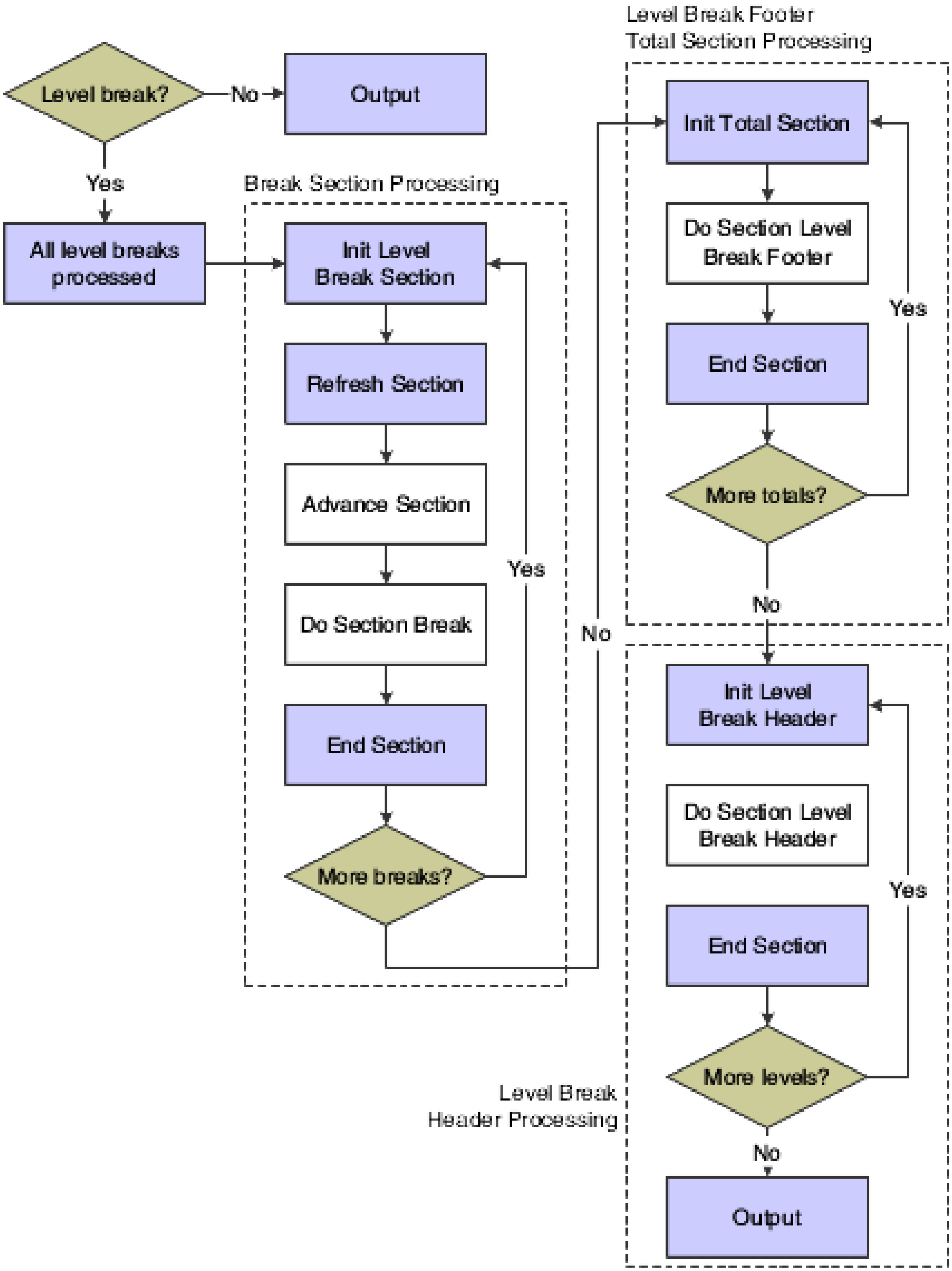
The system identifies a change in the data of a specific field by comparing the current record to the previous record. A level break occurs when the system identifies the change between the records for the fields that you define as level break fields.

You can use level breaks to organize the data of a large report into smaller, logical groups of data that are easier to review. Typically, the system displays a descriptive header before the associated data. For example, you can group address book records by search type. All records with the search type of *E* are grouped together with an appropriate heading. When the value in the search type field changes, a level break occurs and a new grouping begins.

Level breaks can be set up to initiate page breaks, totals, headers, and footers. Only business view fields can be defined as level break fields.

A single business view field can be attached to one level break header and one level break footer in a report template. If more than one level break header or level break footer is attached to a business view field, the batch engine processes only the first level break header or level break footer section and ignores the others.

This diagram illustrates level break logic processing:



Before a level break section is processed, the batch engine issues the Init Break Section event. This event stops processing for the current section and begins processing the level break footer section. After the level break footer section is processed, the batch engine processes the level break header if one is defined.

Level Break Section Events

When a report containing level break sections is processed, the system automatically calls the Init Lvl Brk Footer Section event or the Init Lvl Brk Header Section event. You can attach event rules to these events to control the flow of logic processed by the batch engine.

For example, if you want to create a report that summarizes information by company, you can create a level break footer to organize the data by company. You can define *Company* as a level break field and, when the company value changes, the batch engine will call the level break footer for the detail section.

You can also call business functions, define table input/outputs (table I/O), and include other logic in level break section events.

Init Lvl Brk Footer Section Event

A level break occurs when a new value for the level break field is encountered. Therefore, the new level break value is held in memory at the time that the level break occurs. When the batch system calls the Init Lvl Brk Footer Section event, the values for the previous level are accessed and either summarized or totaled. For example, if you have a level break footer defined to total revenue by company, the level break footer inserts the total following all the associated records that have been processed for a specific company. The Init Lvl Brk Footer Section event is attached to the level break footer section.

If you set up multiple level break footers for a report section, the batch system processes the footers from the lowest level to the highest level. For example, if you define two level break footers, one on business unit and one on company, the system processes the aggregates for business unit first and then the aggregates for the associated company.

When attaching logic to a level break footer, attach it to the End Lvl Brk Footer Section event rather than to the End Section event.

Init Lvl Brk Header Section Event

The Init Lvl Brk Header Section event is called after a level finishes processing and the system has encountered a new value for the level break field. It is also called after the Init Lvl Brk Footer Section event if you have a level break footer defined. The Init Lvl Brk Header Section event locates the header section associated with the previous level and processes the information in the level break header.

For example, if you define a level break header and a level break footer in a report, the system processes the level break footer before the level break header, as shown in the previous *Level break logic processing* diagram.

If you set up multiple level break headers for a report section, the batch system processes the headers from the highest level in the report to the lowest level in the report. For example, if you define two level break headers, one on business unit and one on company, the company level break header processes first, and then the business unit level break header processes.

End Lvl Brk Header and Footer Section Events

After the initialize level break events are processed, the system processes the End Lvl Brk Header Section and End Lvl Brk Footer Section events. These events end the level break process and return to the detail section. You can attach logic to the End Lvl Brk Section event to call a custom section. For example, you might want to print text, such as a disclaimer, after a level break footer. In this case, you can create a custom section that contains the information you want to print and attach the custom section logic to the End Lvl Brk Section event. When you process the report, the disclaimer appears following the level break footer section.

Batch Events

As a batch application processes, the runtime engine pauses at certain points—called *events*—to process logic. You can use events to insert custom logic for processing. The basic flow of events within a section is:

1. Initialize Section
2. Do Section
3. End Section

Some events are processed by the batch engine *only* if the appropriate section type exists. For example, Init Report Header and End Report Header are initialized only if the report includes a report header section. Likewise, Init Lvl Brk Footer and End Lvl Bk Footer are processed only if a level break footer exists.

Do Section Event

The Do Section event is the most commonly used event. It is used most often in columnar and group sections because the Do Section event occurs before any objects are processed.

The Do Section event is invoked *after* the system has assigned new values to objects in the report and immediately *before* the system processes objects within a section.

When the system processes the Do Section event in columnar sections, it processes the column headings first and then fetches the first record. For each object or column in the record, it processes the Init Object, Do Object, and End Object events. After processing the last object, the system calls the After Last Object Printed event and then fetches the next record and repeats the process. When all lines have been fetched from the database, the system processes the End Section event. This basic process is the same for group sections.

If an affected object is a child section, such as in a subsection join, the system invokes the Initialize Section event the first time that the child section is processed. For any subsequent times that the child section is processed, the engine invokes the Refresh Section event.

If an object does not fit on a page, the system invokes the Suspend Object event, which moves the object to the next page.

Report Level Events

This table describes the events that are available at the report level:

Event	Description
Do Initialize Printer	Resolves and validates the printer name. This event is called once per subsystem trigger. The printer can be set using the Initialize Printer system function, which is only valid on this event.
Initialize Report	Resets the global report variable and global event rule variable values if the event contains no event rules in a subsystem; otherwise. preserves global variable values. The Initialize Report event is executed only once per report and is always the first event to be processed. If the report is a subsystem report, the system executes the event rules on this event once as the subsystem begins and before any subsystem triggers are processed.
End Report	Executes once at the end of the report processing and is always the last event to be processed. If the report is a subsystem, the system executes the event rules on this event only after the system processes an End Subsystem trigger and the subsystem is in the process of terminating.

Section Level Events

This table describes the events that are available at the section level:

Event	Description
Advance Section	Occurs each time you perform a fetch from the database. Use this event if you need to perform processing on objects before the fetch. If the section does not have a business view attached, then this event is processed only once.
After Last Object Printed	Occurs after a record is printed to an output file. Use this event to process information after a record has been output.
Before Level Break	Occurs before a level break is processed. Use this event to perform processing after a fetch but before any level breaks are checked.
Do Balance Auditor	Valid only for tabular sections. Use this event for the Drill Down feature.
Do Section	Occurs following an Advance Section event, after values have been assigned to print to a printer or an output file. Occurs before any information for the current record is written to the PDF and before the Do Cell (if tabular cells exist), Do Variable, or Do Constant events. In a tabular section, this event processes after the level break.
Do Tabular Break	Valid only for tabular sections. Occurs when the value changes for a business view field defined as a level break. Use this event to perform processing that requires a change of values in any of the level break fields.
End Break Section	Occurs after a level break finishes processing. Use this event to perform processing immediately after a level break.
End Lvl Brk Footer Section	Occurs after a level break footer. Use this event to do processing immediately after a level break footer.

Event	Description
End Lvl Brk Header Section	Occurs after a level break header. Use this event to do processing immediately after a level break header.
End Section	Occurs after a batch process has completed processing the last set of section values. Use this event to do processing immediately after a section ends. This event is useful for last record and end-of-file procedures.
Init Break Section	Occurs after a level break begins processing. This event initializes a child section that is joined to the parent section on a level break.
Init Lvl Brk Footer Section	Occurs before a level break footer. Use this event to do processing immediately before a level break footer.
Init Lvl Brk Header Section	Occurs before a level break header. Use this event to do processing immediately before a level break header.
Initialize Section	Occurs when a batch process encounters a section for the first time. Use this event to do processing immediately before a section begins. This event is useful for working with global variables or performing other preparatory procedures. For conditional sections, this event is processed each time the section is called. For subsection joins, this event is only processed the first time the child section is processed.
Refresh Section	Occurs subsequent to all but the first time a child section is processed. The first time the batch engine encounters a child section, it issues an initialize section event. Each subsequent time the child section is processed, the batch process uses Refresh Section. At this point, the internal structures and pointers for the child section are established and the batch engine prepares to select a new group of records for the child section. This logic also works for the level break sections. Use this event to set the object values of level-two sections based on the parent section. You can also use this event to reset or modify data selection and data sequencing for the child section.
Suspend Section	Occurs when information exceeds the space available on a page. This event temporarily stops the section processing. Use this event to do processing when a page break occurs.

Page Header Section Level Events

This table describes the events that are available at the page header section level:

Event	Description
Initialize Page Header	Occurs at the beginning of a report, after the report header section is processed and before the page header section processes for the first time. It also processes every time a page break occurs. Use this event to initialize values that cannot be set until after the report header logic executes. This event is similar to Init Section for a normal group, columnar, or tabular section, except that it is processed only for a page header section.
End Page Header	Occurs after the page header finishes processing. Use this event to do processing immediately after a page header.

Event	Description

Page Footer Section Level Events

This table describes the events that are available at the page footer section level:

Event	Description
Initialize Page Footer	Occurs at the beginning of the report, after the report header processes and before the page header section processes for the first time. Use this event to initialize values to be printed in the current page footer section. These assignments typically depend on information already processed on that page.
End Page Footer	Occurs after the page footer finishes processing. Use this event to do processing immediately after a page footer.

Report Header Section Level Events

This table describes the events that are available at the report header section level:

Event	Description
Initialize Report Header	Processes once at the beginning of the report. Use this event to initialize values at the beginning of a report. This event is similar to Init Section for a normal group, columnar, or tabular section, except that it processes only for a report header section.
End Report Header	Occurs after the report header processes. The system processes the page header for the report next. Use this event to do processing immediately after a report header and before a page header.

Report Footer Section Level Events

This table describes the events that are available at the report footer section level:

Event	Description
Initialize Report Footer	Occurs once at the end of a report, after everything else processes and before the report footer prints. Use this event to initialize values to print in the report footer.
End Report Footer	Occurs after the report footer processes. After processing completes, the report terminates. Use this event to do processing immediately after a report footer.

Event	Description

Constant and Variable Events

This table describes the events that are available in constant and variable events:

Event	Description
Do Column Heading	Occurs when the column is initialized. Use this event to populate the column heading based on event rules associated with a business function.
Do Variable and Do Constant	Occurs before the font and color are selected and before the value of the object is translated into a printable string of characters and output to the page. Use this event to do processing after an object has been processed. This is your last opportunity to manipulate the values or display attributes of objects before output.
End Variable and End Constant	Occurs immediately after an object is processed even if the object is invisible or suppressed. Use this event to do processing after an object is processed.
Initialize Variable and Initialize Constant	Occurs before each report object or variable is processed. Use this event to do processing before an object is processed. This event is useful for processing that affects the position of an object because the object's position on the page has not yet been determined.
Suspend Object (constant)	Occurs if an object requires multiple text strings or column headings and if only part of the object fits on a page. This event halts processing of the object until the next page has been started. Use this event to modify the value at the page break. Because the value of the object is already partially processed, this is not a good time to manipulate that value.
Column Inclusion	Valid only for tabular sections. Occurs after each record is fetched from the database. Use this event to perform calculations. Do not use the Column Inclusion event when you are performing calculations between columns (such as when calculating variance) or between variables within a column.
Cell Inclusion	Valid only for tabular sections. Occurs during Do Object after processing calculations for a cell. Use this event to manipulate cell data before displaying it. This event occurs before the Do Variable and Do Constant events. This event occurs during calculations. You can also attach criteria to determine whether the currently fetched data should be included in the cell calculation.

System Functions Within Batch Events

System functions provide you with flexibility and control over how reports are processed. For example, you can use system functions to hide and show objects, hide and show sections, and generate messages.

This table describes the main categories of system functions available in batch applications:

Category	Description
Object	Use to perform actions such as hide and show objects.
Section	Use to perform actions such as hide and show sections or work with totals.
General	Use to perform actions such as work with data selection and data sequencing.
Messaging	Use to perform actions such as send, update, or delete messages.
Workflow	Use to perform actions such as work with processes.
Transaction Processing	Use to begin, commit, or roll back transactions.
Media Objects	Use to work with media objects.

Many system functions are shipped with JD Edwards EnterpriseOne. This table describes some of the commonly used system functions located in the general folder:

System Function	Description
Set Selection Append Flag and Set Sequence Append Flag	Enables you to add, append, or overwrite data selection and data sequencing for a section.
Set User Selection and Set User Sequence	Enables you to define data selection and data sequencing for a section.
Stop Section Processing	Enables you to stop the processing of the current section. The system moves to the next section and begins processing that section. This system function is helpful for performance, especially when there is a large amount of event rule logic that remains to be performed. For example, if no more customers exist with a credit limit over a certain amount, the system stops processing that section and moves to the next section. Stop Section Processing differs from Suppress Section Write. Suppress Section Write suppresses only the current record, which causes the engine to process the next record for the current section.
Hide Object	Enables you to hide objects in group and columnar sections using the Hide Object system function. The system prints a blank line even if the hidden object is the only object on the line because the system has no way of knowing whether the line contains other objects that need to be printed. To keep the system from printing a blank line, place the object in its own conditional section and suppress the printing of the conditional section using the Hide Object system function.

23 Understanding Runtime Processing

Batch Runtime Processing

The term *batch runtime processing* refers to how events and their attached event rules are evaluated at runtime.

Data is stored in memory in an internal data structure, or runtime structure, during runtime processing. Certain fields of the data structure temporarily store data during runtime until it is no longer needed. Then data can be cleared to process another record.

Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA) provides several different field types and event rules that are associated with runtime structures. You need to understand the values that are held in the runtime structure at different points during the processing of a report. To explain the values that are held in runtime structures at a given time, you need to first understand the objects that are available and their two-character, alphabetical code.

Available Objects

Available objects are represented by a two-character, alphabetical code that indicates the source of the data and determines how the data is used in reports and batch processes at runtime.

This table describes the available objects that are defined for batch processing:

Object	Description
BC	Business view columns. Columns that are included in the attached business view. These columns are populated with values from the database when a fetch is performed, and those values are the values that are saved to the database during an add or update.
PO	Processing option values. These values are passed into the application when the batch version is submitted. These processing option values are entered by the user or defined in a particular version of an application.
VA	Event rules variables. Variables that you create in event rules using data dictionary items. They are not manipulated by the system.
SV	System variables. Values that represent environment variables that are made accessible to event rules.
SL	System values. Values that represent constant system values that have been made accessible to event rules.
TV	Text variables. Variables that you create in RDA for use in event rules.
RC	Report constants. Includes column headings in columnar sections and the constant portion of fields in group sections.
RV	Report variables.

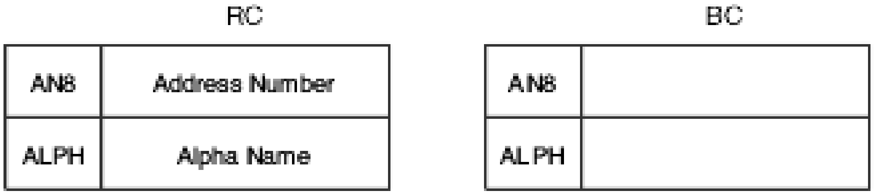
Object	Description
PC	Previous business view columns. Columns that are included in the attached business view. These columns are the previous values for a field rather than the value currently in memory.
PV	Previous report variables.

Typical Event Flow for Group Sections

The runtime engine processes events in a specific order. Some examples are provided in the following sections to help you understand the values that are held in the runtime structures after specific group section events are processed by the system.

Initialize Section

This example illustrates the values that are held in the runtime structures after the system processes the Initialize Section event for the group section of an address book report:



RC

AN8	Address Number
ALPH	Alpha Name

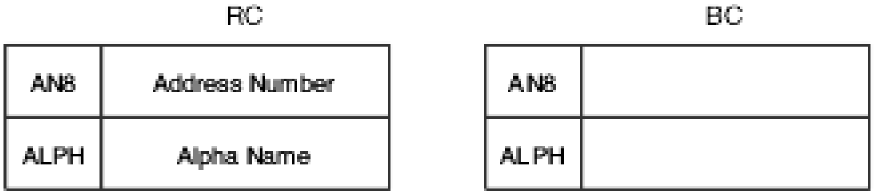
BC

AN8	
ALPH	

Advance Section

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system:

- Initialize Section
- Advance Section



RC

AN8	Address Number
ALPH	Alpha Name

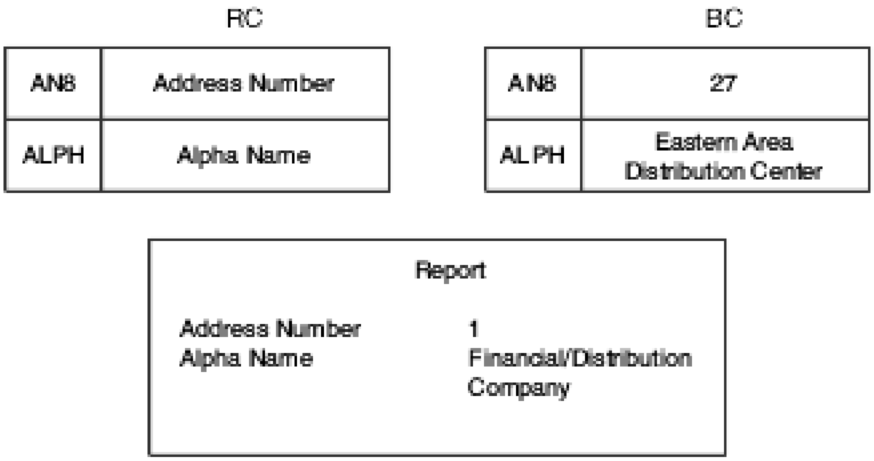
BC

AN8	
ALPH	

Before Level Break

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system:

- Initialize Section
- Advance Section
- Before Level Break



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	27
ALPH	Eastern Area Distribution Center

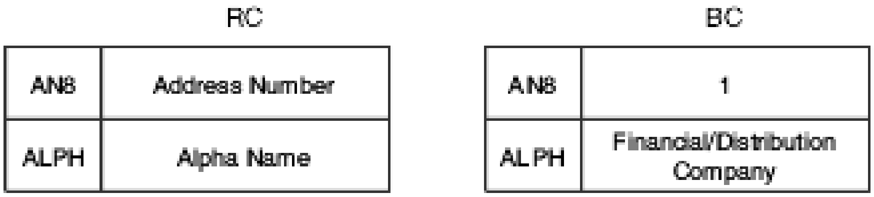
Report

Address Number	1
Alpha Name	Financial/Distribution Company

Do Section

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section



RC

AN8	Address Number
ALPH	Alpha Name

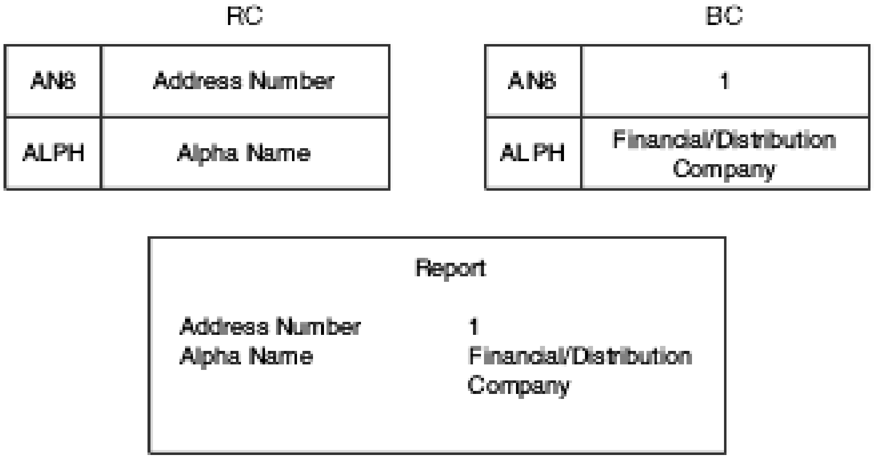
BC

AN8	1
ALPH	Financial/Distribution Company

After Last Object Printed

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	1
ALPH	Financial/Distribution Company

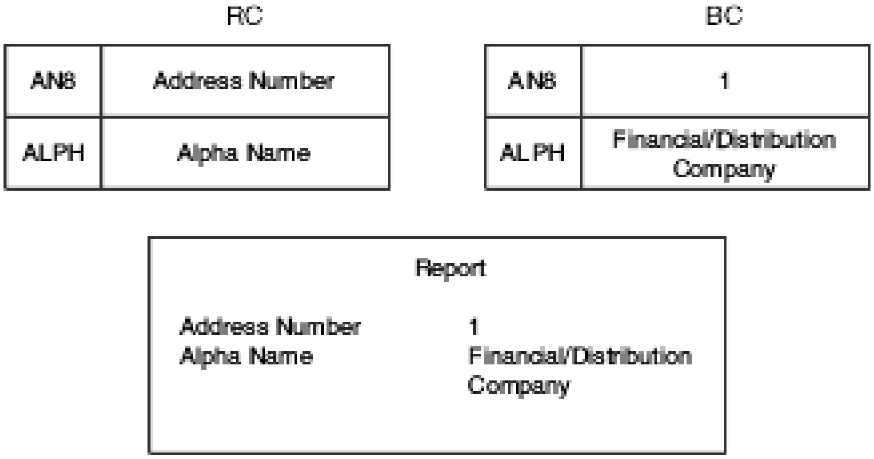
Report

Address Number	1
Alpha Name	Financial/Distribution Company

Advance Section

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed
- Advance Section



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	1
ALPH	Financial/Distribution Company

Report

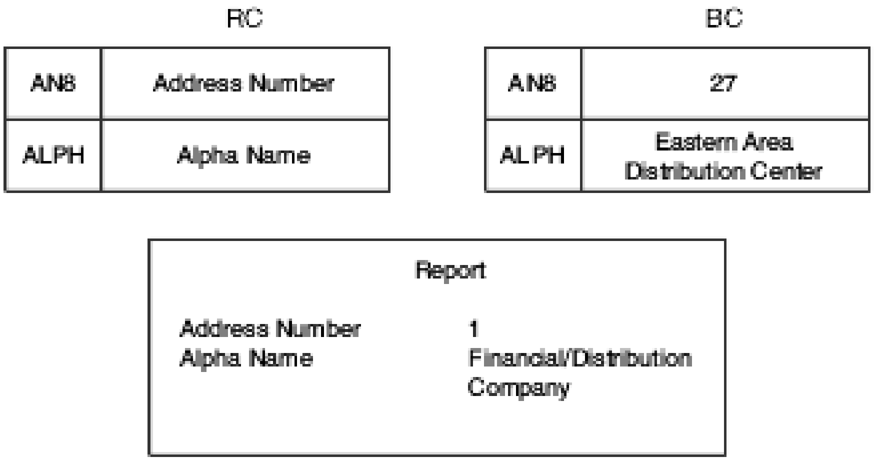
Address Number	1
Alpha Name	Financial/Distribution Company

Before Level Break

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break

- Do Section
- After Last Object Printed
- Advance Section
- Before Level Break



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	27
ALPH	Eastern Area Distribution Center

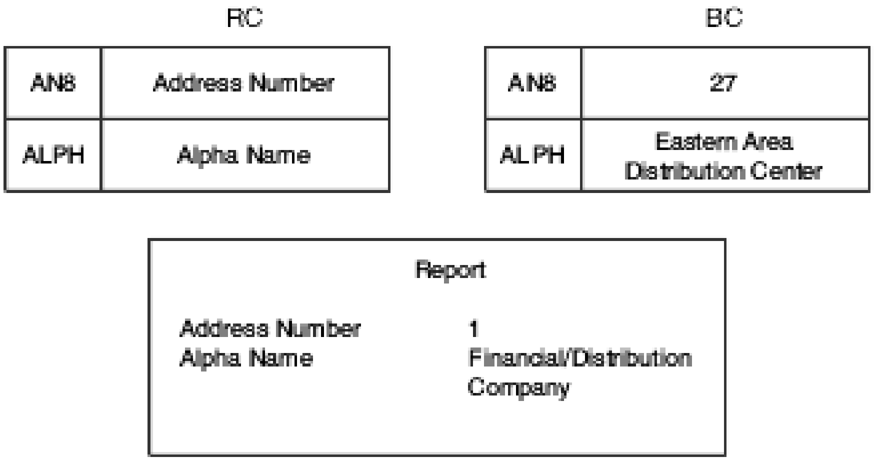
Report

Address Number	1
Alpha Name	Financial/Distribution Company

Do Section

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed
- Advance Section
- Before Level Break
- Do Section



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	27
ALPH	Eastern Area Distribution Center

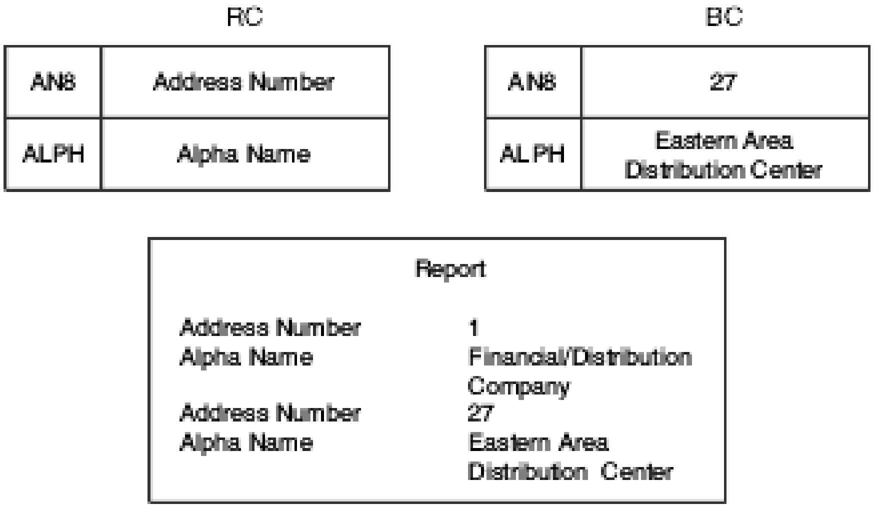
Report

Address Number	1
Alpha Name	Financial/Distribution Company

After Last Object Printed

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	27
ALPH	Eastern Area Distribution Center

Report

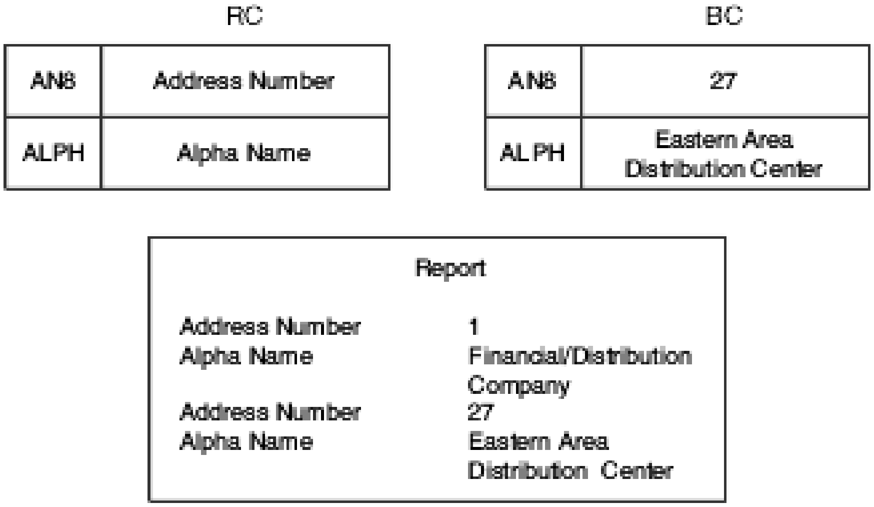
Address Number	1
Alpha Name	Financial/Distribution Company
Address Number	27
Alpha Name	Eastern Area Distribution Center

Advance Section

This example illustrates the values that are held in the runtime structures after these group section events are processed by the system and shows the data that appears in the report:

- Initialize Section
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed
- Advance Section
- Before Level Break
- Do Section
- After Last Object Printed

- Advance Section



RC

AN8	Address Number
ALPH	Alpha Name

BC

AN8	27
ALPH	Eastern Area Distribution Center

Report

Address Number	1
Alpha Name	Financial/Distribution Company
Address Number	27
Alpha Name	Eastern Area Distribution Center

24 Working with Report Interconnects

Understanding Report Interconnects

You can use report interconnects to launch reports and batch applications from:

- Another batch application
- An interactive application

How the system processes the batch application information depends upon the type of processing selected:

- Synchronous

  Synchronous is the default processing method. If you select synchronous processing of the report interconnect, the initiating process waits until the batch application has finished before it continues running the original application.

- Asynchronous

  If you select asynchronous processing, the initiating process starts another process and continues to run the original application. The two processes run separately but at the same time.

Defining Report Interconnects

This section provides an overview of report interconnects in batch applications, lists the prerequisites, and discusses how to create report interconnects.

Understanding Report Interconnects in Batch Applications

You can use report interconnects to automatically run a batch application from another batch application or an interactive application when certain criteria are met. You can use a report data structure along with the report interconnect to pass values between the batch applications.

The batch application that is doing the calling is the primary batch application. The batch application that is being called is the secondary batch application.

Define the report interconnect in the primary batch application. Use the After Last Object Printed event if you want the secondary batch application to be processed *after* the primary batch application is complete.

You must map parameters as part of the report interconnect. These parameters map data from the primary batch application to data in the secondary batch application. As part of the parameter mapping, you must also indicate the direction that the data should flow.

Define the data structure in the secondary batch application. The secondary batch application might not have a business view attached. It can receive its data from the primary batch application based on the criteria that you define. Select data items to include in the data structure, and then assign these fields to the data items that are included in the secondary batch application.

Report interconnects are beneficial when the secondary batch application provides further detail for the records that are included in the primary batch application or when different data in the primary batch application is required for different audiences.

The user runs the primary batch application, and when the stated criteria are met, the system processes the secondary batch application automatically.

Prerequisites

Before you begin creating report interconnects, ensure that you:

- Create two batch application objects.
- Complete the design of the first report template to serve as the primary batch application.
- Complete the design of the secondary report template and define a report data structure.

Creating Report Interconnects

Open the primary report template in Oracle's JD Edwards EnterpriseOne Report Design Aid.

1. From the detail section of the report, access Event Rules Design, and select an event.
2. Click the Report Interconnection option on the toolbar.
3. On the Work with Applications form, select the batch application that you want to process.
4. On the Work with Versions form, select the appropriate version of the batch application that you want to process.
5. In the Available Objects column, select the object that you want to pass, and move it to the Data Structure-Value Column.
6. Indicate the direction of data flow between Value and Data Items.
7. To run the report as a separate process, select the Asynchronously option.
8. To include the report interconnect for transaction processing, select the Include in Transaction option.
9. Click one of these buttons to add notes, and then click OK:
    - Structure Notes
    - Parameter Notes

> **Note:** Event Rules Design displays the Report Interconnection with the Call (UBE <name> Version <name>) statement.

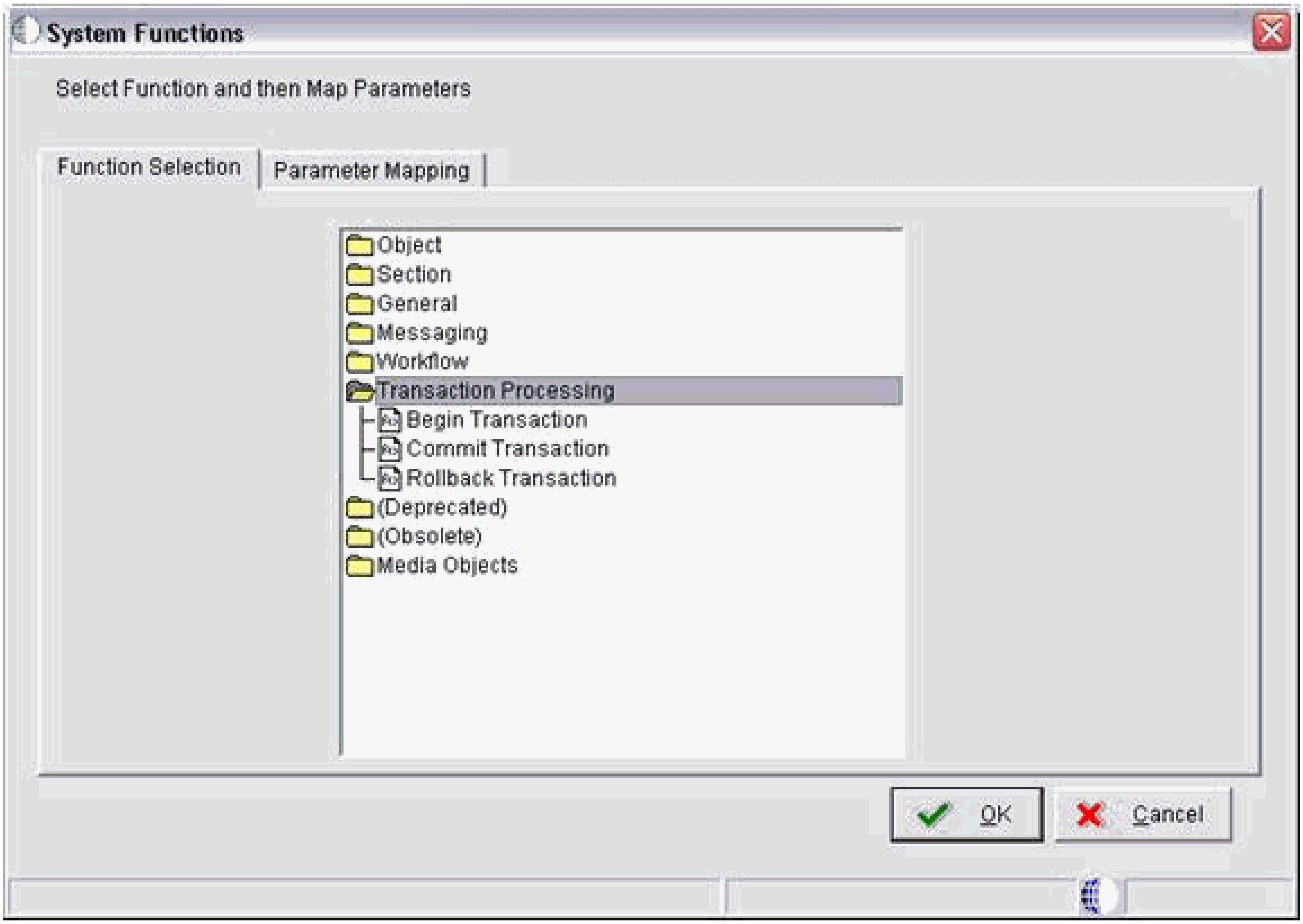
25 Transaction Processing System Functions

Understanding Transaction Processing System Functions

EnterpriseOne uses the following system functions for transaction processing:

- Begin Transaction
- Commit Transaction
- Rollback Transaction

The following screen shows you the transaction processing system functions that display when you choose to attach a system function to an event rule:



To use transaction processing in reports, you must enable the Transaction Processing property.

To enable Transaction Processing:

1. Report Design Aid, open the report for which you want to use transaction processing.
2. Click the File menu, select Report Properties, and then select the Advanced tab.

Ensure that the Transaction Processing option is selected.

Begin Transaction

Call the Begin Transaction system function to start a manual transaction. Calling this system function initiates the beginning of the table updates that are part of the transaction. This system function has one MathNumeric type parameter named Transaction ID. You can use a Report Level Math Numeric variable to pass into the Transaction ID parameter. Even though you must pass in a value for Transaction ID, the value is not currently being used by the system. This is because at any one time, only one transaction can be started from Event Rules.

After calling the "Begin Transaction" system function, the report can use Table I/O statements or Business Functions to conduct database table operations.

To include the Table I/O statements in the current transaction.

1. Click the system function you have assigned to the event rule.
2. Click the Table I/O icon.
3. On Insert Table IO, in the Advanced Operations section, select Open, and then click Next.
4. On Data Source, click the Advanced Options button.
5. On Advanced Options, click the Include in Transaction option, and then OK.

Commit Transaction

Call the Commit Transaction system function to commit all the database operations started since the Begin Transaction call. Similar to Begin Transaction, this system function takes in a Transaction ID parameter. You can use the same variable as the one used in Begin Transaction system function.

EnterpriseOne will commit all database operations started between the Begin Transaction call and the Commit Transaction call. If the Commit Transaction fails, EnterpriseOne automatically rolls back the database operations so that none of the database operations will be committed. There is no need for Event Rules (ER) to call Rollback Transaction after Commit Transaction.

You need to carefully define the location of the Begin Transaction and Commit Transaction calls. The transaction boundary must be just big enough to include all relevant database operations and not too large that it blocks other code from accessing the same database records.

The example below shows how to use transaction processing through Event Rules:

```
// Start Example
Begin Transaction (transaction ID)

Table IO_Open Table 1

Table IO_Insert Table 1

Table IO_Open Table 2

Table IO_Open Table 2
```

```
Commit Transaction (transaction ID)
// End Example
```

Rollback Transaction

Call this system function to cancel all the database operations started since the Begin Transaction and the End Transaction. This system function takes a MathNumeric type parameter called Transaction ID. Use the same variable as used in Begin Transaction. The EnterpriseOne will end the transaction without committing any database operations started between the Begin Transaction and Rollback Transaction calls. You should only use this system function when absolutely necessary.

System Variable - SV TP Commit Status

This system variable tells you the status of the Commit Transaction or Rollback Transaction call. It may contain one of the four valid values:

CO TP_ACTION_FAIL

If SV TP Commit Status is equal to CO TP_ACTION_FAIL, then the last transaction action has failed. The last transaction action can be either Commit Transaction or Rollback Transaction. If the last transaction is Commit Transaction then the Commit Transaction has failed and no database operation has been committed. If the last transaction is Rollback Transaction, then the rollback operation has failed. No database records have been committed.

CO TP_ACTION_SUCCESS

If SV TP Commit Status is equal to CO TP_ACTION_SUCCESS, then the last transaction action has succeeded. The last transaction action can be either Commit Transaction or Rollback Transaction. If the last transaction is Commit Transaction, then the commit has succeeded and all database operations have been committed. If the last transaction is Rollback Transaction, then the rollback operation has succeeded.

CO TP_IN_TRANSACTION

If SV TP Commit Status is equal to CO TP_IN_TRANSACTION, then the transaction has been started, but no Commit Transaction or Rollback Transaction system function has been called yet.

CO TP_NO_TRANSACTION

If SV TP Commit Status is equal to CO TP_NO_TRANSACTION, then the no transaction has not been started or completed by the Transaction Processing system functions.

Event Rules Sample Using System Variable TP_Commit_Status

```
Begin Transaction (transaction ID)

Table IO_Open Table 1

Table IO_Insert Table 1

Table IO_Open Table 2

Table IO_Open Table 2

Business function inserts into Table 3 in a separate transaction.

Commit Transaction (transaction ID)
```

```
if (SV TP_Commit_Status is equal to CO TP_ACTION_FAIL)
{
 // this means the transaction failed (system has rolled it back)
// call another Business Function to roll back Table 3 since Table 3 Insert
// is done outside of current transaction
}
else
{
 // notify user transaction successful
```

26 Setting Up Business Views as Favorites

Understanding Business View Favorites

Business view favorites organize your most frequently used business views. During the report creation process, the Director presents the business views that are set up as favorites. You can quickly access and select from these business views for use in the report section.

You create business view favorites using the Oracle's JD Edwards EnterpriseOne Favorites (P9100) application, which you can access from the Microsoft Windows client or the web client.

Business view favorites enable you to logically organize the business views that you designate as favorites. You can attach notes to business view favorites to include information about the business view. For example, you might include text that describes when to use a particular business view and the name of the business view. For custom business views, you might want to include the table names that are used to create the business view.

You can set up business view favorites to include business views that you use often. For a global company, you can translate the notes and the descriptions of business view favorites to multiple languages.

Setting Up Business Views as Favorites

This section provides an overview of favorites folders and subfolders and discusses how to:

- Add favorites folders.
- Add favorites subfolders.
- Add business views to favorites folders and subfolders.
- Add notes for favorites, folders, and subfolders.
- Modify and delete notes.
- Set up favorites description translations.
- View favorites with alternative descriptions.

Understanding Favorites Folders and Subfolders

Business view favorites are organized in a tree structure using folders and subfolders. You can use these folders in any way that enables you to easily locate frequently used business views. For example, you can name a business view folder using your name. Then you can create subfolders under the name folder for each category of business view that you use often. You can name subfolders after the system that uses the fields that are included in the business views or use another method that better meets your business processes.

Add notes to folders and subfolders when you need to provide information about the business views that are contained within them. You can also use category codes to further organize your business view favorites.

Forms Used to Set Up Business Views as Favorites

Form Name	FormID	Navigation	Usage
Work With Favorites	W91100A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Favorites	Add, modify, and delete business view favorites folders and subfolders.
Object Folder Revisions	W91100B	Click Add on the Work With Favorites form.	Add favorite folders Enter the folder name, parent folder name (if appropriate), description, owner, and descriptive category codes.
Favorites Revisions	W91100C	Select a business view favorites folder or subfolder on the Work With Favorites form, and select Revise Favorites from the Row menu.	Add business views to the business view favorites folders and subfolders.
Notes Revisions	W91100E	Select a business view favorites folder, subfolder, or business view on the Work With Favorites form, and Select Note Revisions from the Row menu.	Add, modify, and delete information regarding the business view favorites folders, subfolders, and business views.
Work With Favorites Description Translation	W91100F	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Favorites Description Translation	Locate an alternative language in which to display business view favorites.
Revise Favorites Description Translation	W91100G	Select a language from the Work With Favorites Description Translation form.	Locate business view favorites, and enter an alternative description in the selected language.

Adding Favorites Folders

Access the Object Folder Revisions form.

Folder
Enter the name of the folder.

Description

Enter a meaningful description of the folder. The description appears on the Work With Favorites form and in Report Design Aid (RDA).

Folder Owner
Enter the address number of the person for whom the business view favorites are defined.

Category Codes 1–6
Enter user-defined values to further categorize the folder.

Adding Favorites Subfolders

From the Work With Favorites form, click a business view favorites folder, and select Add Subfolders from the Row menu.

Adding Business Views to Favorites Folders and Subfolders

Access the Favorites Revision form.

Object Name
Enter the name of the business view favorites folder.

Description
Displays the description of the business view. The system populates this field when you press to move out of the Object Name field. The description appears on the Work With Favorites form and in RDA.

Adding Notes for Favorites, Folders, and Subfolders

Access the Notes Revisions form

Modifying and Deleting Notes

Access the Notes Revisions form.

1. Modify text in the workspace, and click OK.
2. Delete text in the workspace, and click OK.

Setting Up Favorites Description Translations

You must complete two tasks to set up business view favorites in other languages:

- Define the translation language.
- Enter the business view favorite in the translation language.

Defining the Translation Language

Access the Work With Favorites Description Translation form.

Language Preference

Enter the user-defined code (UDC) value for the language in which the business view favorites are written. This is typically the domestic language that is defined for the system.

To Language

Select the alternate language in which you want the business view favorites to be written.

Entering the Business View Favorite in the Translation Language

Access the Revise Favorites Description Translation form.

Language

Displays the UDC value that was entered in the Language Preference field on the Work With Favorites Description Translation form.

To Language

Displays the language that was selected in the To Language field on the Work With Favorites Description Translation form.

Skip To Favorite

Enter the name of the business view favorites folder, subfolder, or business view for which you want to create an alternate description.

Alternative Description

Enter the description of the business view favorites folder, subfolder, or business view as it should appear in the selected language.

Viewing Favorites with Alternative Descriptions

Access the Work With Favorites form.

1. In the Language field in the upper right-hand side of the form, enter the language in which you added an alternative description.

   You can also use the visual assist to search and select a language.
2. Click Find.

   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.

27 Using Date Titles in Financial Reports

Understanding Date Titles

Date titles are used to indicate the time period covered by the data in the report. This table provides examples of several date titles that are included in Oracle's JD Edwards EnterpriseOne:

Date Title	Example
A (As of)	As of 03/31/07
B (Balance sheet)	As of March 31, 2007
P (Profit and Loss)	For the Three Months Ending March 31, 2007
S (Single period)	For the Month Ending March 31, 2007

The User Defined Date Title business function (B8300007) uses the company number to determine the fiscal year. If the user leaves the processing option values for the reporting month and year blank, the system uses the company number to determine the default reporting period.

Using Date Titles in Financial Reports

This section provides an overview of customizing date titles, lists the prerequisites, and discusses how to:

- Define custom date titles.
- Preview date titles.
- Add variables to financial reports to display date titles.
- Create event rules in financial reports to display date titles.

Understanding Customizing Date Titles

For reports that include financial data, a date title in the page header makes the report data more meaningful. JD Edwards EnterpriseOne includes several commonly used date titles. While these predefined date titles enable you to add a date title quickly, they might not be specific to your reporting needs. Therefore, you can add a custom company-specific date title using the Oracle's JD Edwards EnterpriseOne Date Titles (P83100) application.

Call the User Defined Date Title business function (B8300007) to add an existing date title to a report. In addition, you can create the same date title in multiple languages. The date title parameters are stored in the Date Title (F83100) table.

When creating a custom date title you must enter this information:

- Date title type

  A single unique identifier. For a profit and loss date title use the letter *P* as the title type.
- Description

  A description of the date title. For a profit and loss date title enter *Profit and Loss* as the description.
- Language

  Identifies the language. Leaving this field blank selects the domestic language.

As part of the date title creation, the system presents you with 14 fields for defining the elements of the title. Use these element fields to indicate how the date title should appear in reports. If you want a comma to appear in the date title, you must enter a comma in the appropriate element field.

The maximum length of a date title is 100 characters. The maximum number of elements is 14. Each element can be either a literal text string or a text substitution parameter. The system stores these parameters in user defined code 83/TS. The system populates the parameters at runtime, typically through a processing option.

This table describes the text substitution parameters that are available. The predefined elements from which you can select are:

Pre-Defined Elements	Description
@1: Period Name	Fetched by the system from the F83110 table. This name is generally the name of the month that is associated with the period. The system populates the values in the table when you define the name of the period for a fiscal date pattern.
@2: Day Period Ends	Fetched by the system from the Date Fiscal Patterns (F0008) table. For example, this parameter returns 31 for a period that ends on the 31st day of the month.
@3: Century and Year	Fetched by the system from a processing option, but the value might be determined by the company that is selected for the report.
@4: Year	Fetched by the system from a processing option, but the value might be determined by the company that is selected for the report.
@5: Text for Period Number	Fetched by the system from the user defined code (UDC) 83/PT. This UDC stores text for the number of periods through the current period. For example, Period 2 retrieves two periods.
@6: Date (12/31/05)	Fetched by the system from a processing option, but the value might be determined by the company that is selected for the report.

You can use as little as two of these element fields. For example, the elements of the *As Of Date* date title are defined as:

- Element 1 - **As of**
- Element 2 - **@6**

The **As of** is the text that appears on the report. The **@6** fetches the date. An example of this date title, as it appears in the page header of a financial report, might be *As of 12/31/05.*

Select Period Text from the Form menu on the Date Title Revisions form to revise the UDC values for the Text for Period Number (@5) element.

To use date titles in reports, call a date title business function in event rules. In the business function data structure, map the appropriate date title type, the processing option fields, and other relevant fields, such as Company. You must assign values to each data item that is included in the data structure. All directional arrows are defined in the data structure.

Prerequisites

Before you use date titles in financial reports, ensure that you:

- Create a batch application object.
- Complete the design of a financial report template to which you can add a date title.

Forms Used to Define Custom Date Titles

Form Name	FormID	Navigation	Usage
Work With Date Titles	W83100A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Date Titles	Add, modify, and delete date titles.
Date Title Revisions	W83100B	Click Add on the Work With Date Titles form.	Enter the date title type, description, language, and elements.
Date Title Preview	W83100D	Select a date title on the Work With Date Titles and select Preview from the Row menu.	Review and modify date title information.

Defining Custom Date Titles

Access the Date Title Revisions form.

Date Title Type
Enter a unique single character to identify the date title.

Description
Enter text that describes the date title using 30 characters or less.

Language
Select a UDC (01|LP) that indicates the language in which the date title should be displayed.

Elements

Enter components of the date title. For example, **For the six months ending June 2005**, use UDCs (83|TS) for text substituted values such as the *six*, the *June* and the *2005* in the example.

Previewing Date Titles

Access the Date Title Preview form.

Date Title Type
Edit this field if required. The system populates this field based on the information on the Work With Date Titles form.

Language
Edit this field if required. The system populates this field based on the information on the Work With Date Titles form.

Company
Enter a company address book number in this field.

Period Number
Enter a number to indicate the current accounting period. If you leave this field blank, the system retrieves the value from the Financial Reporting Date in the company's application.

Fiscal Year
Enter a two-digit number to indicate the current fiscal year. If you leave this field blank, the system retrieves the value from the Financial Reporting Date in the company's application.

Preview Date Title

On the Date Title Preview form, select Run Preview from the Form menu to preview the date title in the same format as it appears in the report. If the system does not find language-specific versions of the date title, the date title appears in the default language.

> **Note:** If the date title is over 100 characters, you will receive an error message and the date title will be truncated to fit the available space.

Adding Variables to Financial Reports to Display Date Titles

Open a financial report template in Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

1. Click the page header section in the report template and select Alpha Variable from the Insert menu.
2. Position the field in the page header.
3. Double-click the alpha variable field that you just inserted.
4. On the Variable Properties form, enter a name in the Variable Name field on the Description tab.

   Use a name that you can easily identify in Event Rules Design, such as Date Title.
5. Select the Display tab and set the Justification to Center.
6. Set the Display Length to **100.**

   If the date title is over 100 characters, the system truncates the date title to fit the available space.
7. Select the Advanced tab, select the Global Variable option and click OK.

   Selecting this option makes the variable available in event rules for all sections of the report.

8. Align the object with the other objects in the section and save the report template.

Creating Event Rules in Financial Reports to Display Date Titles

Open a financial report template in RDA to which you have added a variable for displaying data titles.

1. Click the tabular section of the financial report template and select Event Rules from the Edit menu.
2. Select the Initialize Section event from the events drop-down list box.
3. From the Insert menu, select Business Function.
4. On the Business Function Search form, enter **B8300007** in the Source Module field of the QBE line and press Enter or click Find.
5. Click Select to select the business function.
6. On the Business Functions form, click in the Value field next to the data item **cDateTitleType** to map the data structure parameters.
7. Double-click Literal in the Available Objects list.
8. On the Single value form, click the visual assist in the Date Title Type field.
9. On the Date Title Search form, select a value, and click Select.
10. On the Single value form, click OK.
11. Click in the Value field next to the data item **szLanguagePreference.**
12. To print the date title in the user's language, double-click SL LanguagePreference in the Available Objects list.

    SL LanguagePreference is the system value for language preference in the user profile.
13. To print the date title in a specific language, regardless of the preference of the user running the report, double-click Literal in the Available Objects list.
14. On the Single value form, click the visual assist in the Language field.
15. Select a value and click Select.
16. On the Single value form, click OK.
17. Click in the Value field next to the data item **szCompany.**
18. Double-click Literal in the Available Objects list.
19. On the Single value form, enter the company number in the Company field and click OK.

    The company that you select determines the fiscal date pattern.
20. Click in the Value field next to the data item **mnPOPeriodNumber**, and then double-click PO PeriodNoGeneralLedger in the Available Objects list.

    This is the period number that is defined in the Financial Reports processing options (T83PO). The processing option appears automatically at runtime to prompt the user for a value.
21. Click in the Value field next to the data item **szPOFiscalYear**, and then double-click PO szFiscalYear in the Available Objects list.

    This is the fiscal year that is defined in the Financial Reports processing options (T83PO).
22. Click in the Value field next to the data item **szDateTitle**, and then double-click the report variable name beginning with RV in the Available Objects list.

    This is the name that you assigned to the alpha variable that you inserted into the Page Header.
23. On the Business Functions form, click OK.
24. On the Event Rules Design form, click the check mark to save and return to RDA.

Assigning Accounting Periods to Column Headings

This section provides an overview of accounting periods and discusses how to define column headings for accounting periods.

Understanding Accounting Periods

You must define each accounting period in the Oracle's JD Edwards EnterpriseOne Column Headings (P83110) application or the Oracle's JD Edwards EnterpriseOne 52 Period Column Headings (P83110B) application, according to your business requirements. You can access both column heading applications from both the Microsoft Windows client and the web client.

Each fiscal date pattern type is assigned a name for each period. For example, period six might be equal to June. The system uses the name that is assigned to a period in the Date Title and in the column headings of smart fields. Each fiscal date pattern type can have period names to accommodate company-specific fiscal date patterns. For example, a fiscal date pattern for a fiscal year that begins in October has a column heading of October for period 1. A fiscal year that begins in January has a column heading of January for period 1.

Forms Used to Assign Accounting Periods to Column Headings

Form Name	FormID	Navigation	Usage
Work With Column Headings	W83110A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Column Headings	Add, modify, and delete accounting period column headings.
Column Heading Revisions	W83110B	Click Add on the Work With Column Headings form.	Enter fiscal date pattern, language, and period names.

Defining Column Headings for Accounting Periods

Access the Column Heading Revisions form.

Fiscal Date Pattern
Select a UDC (H00|DP) to identify a date pattern. Codes must be defined for this use.

Language
Select a UDC (01|LP) to indicate the language in which the column headings should appear.

Period 1–7 and Period 8–14
Enter column heading names for each accounting period. For example, January, February, and so on.

28 Setting Up Processing Option Templates

Understanding Processing Option Templates

Processing options designate parameters that are used by reports. When reports are processed, the processing option template passes values that are entered by the user. These values are used by the event rules that are attached to the report to perform customized processing.

Processing option templates are included in Oracle's JD Edwards EnterpriseOne. You can also create custom processing option templates to meet your business needs.

You can use processing options to:

- Control how reports process data.
- Set up default values.
- Customize batch versions for different companies or even different users.
- Control the format of reports.
- Control page breaks for reports.
- Control totaling for reports.

Processing option templates can contain one or multiple parameters. You can add tabs, with descriptive names, to the processing option template to categorize the parameters. For example, you can name a tab *Print* to describe parameters that affect how a report prints or which printer is used.

At runtime, processing option templates display the set of tabs which are referred to as *pages*. When you select a tab, the page changes to display the set of processing options for that category.

You can add comments to processing options. The comment should describe the purpose of the available parameters and any information that is required by the user. For example, Use this processing option to indicate the preferred format of labels for printing.

Each processing option page displays parameters that are accompanied by a description. The description often includes valid values when appropriate. Add data items to the processing option page to create the parameters. You can change the description of the data item to better describe the purpose of the parameter. For example, **Enter 1 to print single column labels and 2 to print two-column labels.**

Each data item that you include on the processing option template includes a member name. This name is located on the Properties form of each parameter. You want to modify the member name of generic fields to represent their specific purpose. Modify the member name to be a descriptive name that you will recognize in Event Rules Design. The recommended naming convention for member names is:

- Add the Hungarian Notation at the beginning of the name.
- Append the data item alias at the end of the name, preceded by an underline.
- Do not include spaces in the name.

For example, the Add [Y/N] data item is often used in processing options for entering a *yes* or *no* value. The alias for the Add [Y/N] data item is A, and the field is defined as a character data type. With that in mind, you might modify the member name to read *cDisplayCommentsColumn_A*.

Note:

- *"Using Processing Options" in the JD Edwards EnterpriseOne Tools Data Structure Design Guide* .

Designing and Using Processing Option Templates in Reports

This section provides an overview of processing option templates in reports, lists the prerequisites, and discusses how to:

- Create data structure objects.
- Create processing option templates.
- Add tabs to processing option templates.
- Attach processing option templates to reports.

Understanding Processing Option Templates in Reports

To use a processing option with a report, you must attach the processing option template to the report template. Attach the processing option template by selecting Select Processing Options from the File menu. You can then search for and select the appropriate processing option template from the Select Processing Option Template form.

After you attach the processing option template to the report template, you must create event rules to tell the system how to process the processing option values, and under what conditions. You should include all valid values in the event rules. You should also include a default scenario in case the user leaves the processing option blank.

Processing option templates are report features so they are always attached to the report template. Processing option templates are reusable objects so you can attach the same processing option template to multiple reports. Processing option values, however, are stored with the batch version. This enables you to assign a set of processing option values to a particular batch version. For example, you can have two versions of a report that prints labels; one version with the processing option values set to print one-column labels, and one version set to print two-column labels.

When the report is launched, the system prompts the user to enter processing option values. However, even if a report template has a processing option template attached, the system might launch the report with no prompting. This is called a *blind execution*. The processing option values for a blind execution are predefined and cannot be modified by the user.

This overview outlines the process for creating and using processing option templates in a report:

1. Create processing options by building a list of parameters called a *template*.
2. Attach the processing option template to a report template and create event rules in the appropriate report section to process the processing option values.
3. Create a batch version of the report template and select one of these options to indicate how the system should handle the processing options at runtime:
   - Prompt for Values

     The processing options appear, enabling the user to enter values.
   - Blind Execution

The report runs using a set of predefined processing option values.

Prerequisites

Before you design and use processing option templates in reports, ensure that you:

- Create a batch application object.
- Complete the design of a report template to which you can attach a processing option template.

Creating Data Structure Objects

Access Oracle's JD Edwards EnterpriseOne Object Management Workbench.

1. Select the project in which you want to add a new processing option template and click Add.
2. On the Add EnterpriseOne Object to the Project form, select the Data Structure option and click OK.
3. On the Add Object form, enter the name of the processing option template in the Object Name field using the recommended naming conventions.
4. Enter a brief description of the processing option template in the Description field
5. Enter a client reserved product code (55–59) in the Product Code field.
6. Enter the system code that relates to the report data that this processing option will affect in the Product System Code field.
7. Enter **360** in the Object Use field to indicate that the object is a data structure.

   The object use value should reflect the object that you are creating. You can create categories of data structures to select from the Object Use list. Click the visual assist in the Object Use field. On the Select User Defined Code form, select Revisions from the Form menu. On the Work With User Defined Codes form, click Add. On the User Defined Codes form, scroll to the bottom of the list and enter a new code and description in the blank row of the detail area.
8. Under Type, select Processing Option Template and click OK.

   The tool that the system presents for developing the object is dependent on the data structure type that you select. If you select Regular Data Structure, the system presents a much different form than if you select Processing Option Template.
9. On the Processing Option Design form, select the Design Tools tab, and click Start the Processing Option Design Aid.

Creating Processing Option Templates

Access Oracle's JD Edwards EnterpriseOne Processing Option Design Aid.

1. On the Processing Options Design form, right-click <New Tab> and select Current Tab Properties from the pop-up menu.
2. On the Tab Properties form, complete these fields, and click OK:

   - Short Name
   - Long Name

3. Click A on the toolbar and then click the page of the processing option template to add a comment box.
4. Double-click the comment box, highlight Comment Text, and enter a brief description of the processing option parameters.

   > **Note:** You can highlight the comment text and type over it or you can backspace to remove the text, but you cannot use Delete to remove the text.

5. On the Data Dictionary Browser form, enter a value on the QBE line to locate a required data item and then press Enter.

   You can also click the Search button that is located directly beneath the Data Dictionary Browser heading to search through all data items.
6. Drag the required data items to the page of the processing option template.

   If you double-click the data item, it appears on the selected processing option page.
7. To reposition a data item on the processing option page, click the data item and drag it to another location.

   The system automatically adjusts the size and position of data items to fit the width of the tab.
8. Double-click the text portion of the item to modify the description.

   > **Note:** You can highlight the text and type over it or you can backspace to remove the text, but you cannot use Delete to remove the text.

9. Right-click the data item field and select Properties from the pop-up menu.
10. On the General tab of the JDE.DataItem Properties form, change the Member Name field.
11. Select the Help Override Data Item tab, modify the Data Item Help Override Name field, and click OK.
12. Save the processing option template and exit JD Edwards EnterpriseOne Processing Option Design Aid.

Adding Tabs to Processing Option Templates

Open a processing option template in JD Edwards EnterpriseOne Processing Option Design Aid.

1. Right-click the current tab and select New Tab from the pop-up menu.
2. On the Tab Properties form, complete these fields, and click OK:

   - Short Name
   - Long Name
3. Add comments and data fields as appropriate.
4. When the new tab is complete, select Test from the Edit menu to test the processing option.

   In test mode, you can click the visual assist for a data item to verify that you have selected the correct data item.
5. Save the processing option template and exit JD Edwards EnterpriseOne Processing Option Design Aid.

Attaching Processing Option Templates to Reports

Open a report template to which you can attach a processing option in Oracle's JD Edwards EnterpriseOne Report Design Aid.

1. From the File menu, click Select Processing Options.

2. On the Select Processing Option Template form, use the QBE line to search for and select the processing option template that you want to use and click OK.

   A check mark appears next to Select Processing Options on the File menu, indicating that a processing option template is attached to this report.
3. Verify that the processing option template is attached to the report by selecting Report Properties from the File menu.

   The Processing Options field on the Report Properties tab indicates the processing option template that is attached to the report.
4. Add logic in Event Rules Design to indicate to the system how each valid value entered into the processing option template should be processed.
5. To remove an existing processing option template, click Select Processing Options from the File menu.
6. On the Select Processing Option Template form, search for and select the processing option template and click Remove.

29 Working with Subsystem Jobs

Understanding Subsystem Jobs

Within Oracle's JD Edwards EnterpriseOne, subsystem jobs are batch processes that continually run independent of, but asynchronously with, JD Edwards EnterpriseOne applications. These subsystem jobs function with the system's logical process or queue defined for the server platform. You can configure JD Edwards EnterpriseOne to use one or more subsystems.

Use subsystem jobs to:

- Off-load processor resources.
- Protect server processes.
- Perform repetitive and frequent processes to maximize output throughput.

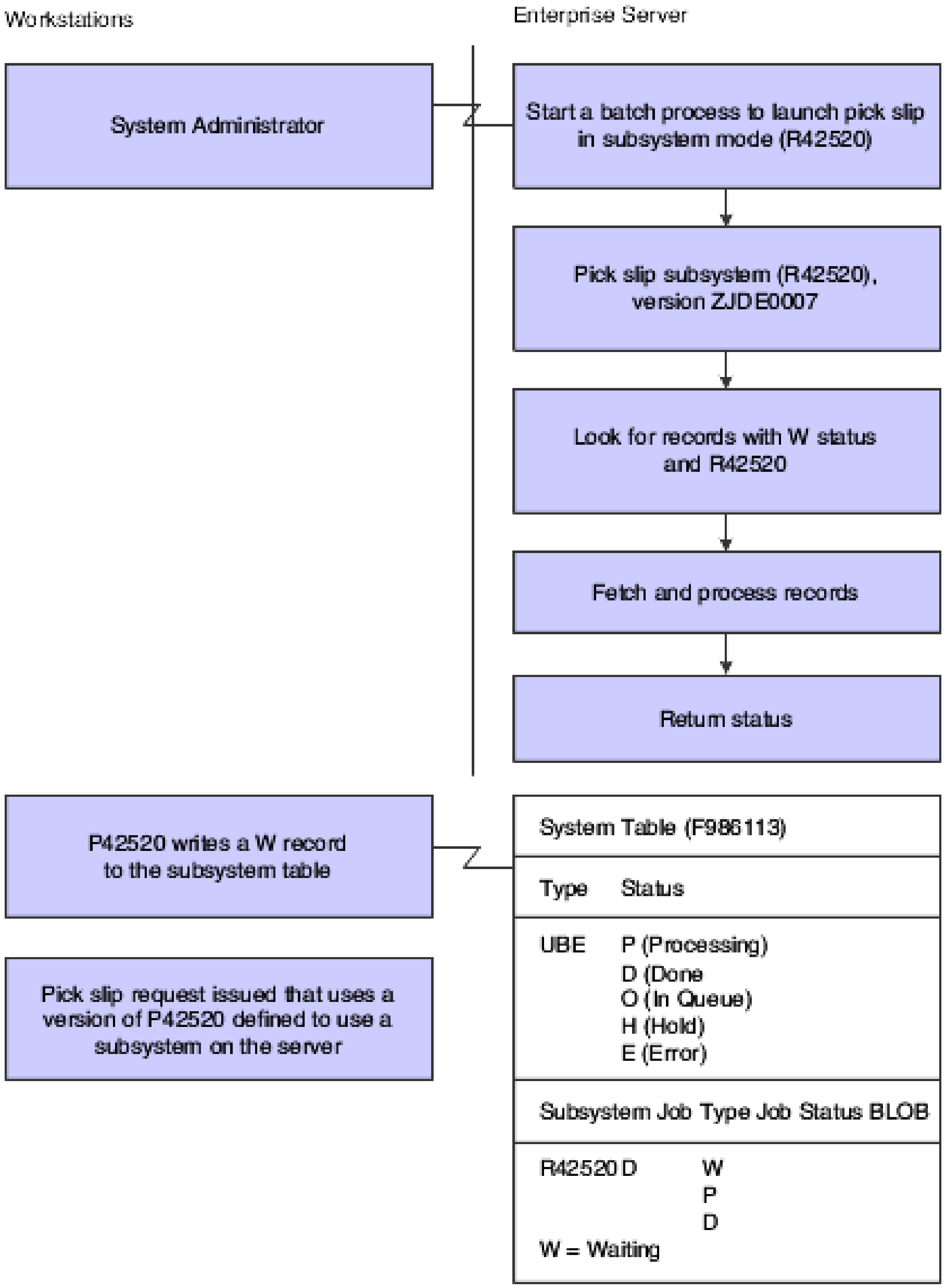
Examples of applications that are suited for subsystem processing include:

- Logistics Warehousing
- Inventory
- Sales Order Processing

For example, you can execute the Sales Order Entry application on a workstation and automatically print pick slips when all orders are entered. If you are using a version of pick slips that has the subsystem job function enabled, the pick slip request is routed to and processed by the subsystem job on the defined enterprise server. As a result, no additional processing resources are required of the workstation.

When a JD Edwards EnterpriseOne application issues a request for a job to run as a subsystem job, it places a record in the Subsystem Job Master (F986113) table. This record is identified by a subsystem job name and version, and contains status and operational indicators. Embedded in the record is key information that enables the subsystem to process the record without additional interaction with the requesting application. Key information includes the values for the processing options and the values for the report interconnect data structure. The continuously running subsystem monitors this table for records. If the subsystem finds a record with the appropriate status indicators for the specific report and version being run, it processes the record and updates the status accordingly.

This diagram illustrates how the system processes a subsystem job:


Workstations
Enterprise Server
System Administrator
Start a batch process to launch pick slip in subsystem mode (R42520)
Pick slip subsystem (R42520), version ZJDE0007
Look for records with W status and R42520
Fetch and process records
Return status
P42520 writes a W record to the subsystem table
Pick slip request issued that uses a version of P42520 defined to use a subsystem on the server
System Table (F986113)
Type Status
UBE
P (Processing)
D (Done
O (In Queue)
H (Hold)
E (Error)
Subsystem Job Type Job Status BLOB
R42520 D
W
P
D
W = Waiting

Defining Subsystem Jobs

This section provides overviews of subsystem job definitions and the process of adding records to the subsystem table using an API, lists the prerequisites, and discusses how to:

- Define reports as subsystem jobs.
- Add records to the subsystem table.

Understanding Subsystem Job Definitions

Subsystem jobs are jobs that continuously process records from a data queue. This type of job runs until you request termination.

The tasks performed by subsystem jobs are:

- Reading records one at a time from a subsystem table.
- Retrieving information for the specific record.
- Running the batch engine for each record.

You run a subsystem job in the same way that you run a regular batch job; no difference exists between running a subsystem job and running a batch job. Before processing, the system ensures that limits for the subsystem job on the defined server have not been exceeded. If exceeded, the batch engine does not process the subsystem job. At the end of the records, instead of ending the job, subsystem jobs sleep for a defined period and then retrieve the information for the next record. For each subsystem job, multiple records can exist in the subsystem table.

Understanding the Process of Adding Records to the Subsystem Table Using an API

You can use an API to add records to the subsystem table to enable the subsystem job to perform batch processes. To add records to the subsystem table:

1. Create a business function using Microsoft Visual C++.
2. In Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA), create a report data structure in the report template.

   Add the appropriate data items to the report data structure.
3. Generate a header file in the report template.

   The header file takes the form of `Report_Name.h` and is located in the $environ\include subdirectory under the folder.

   E812

   This example illustrates the subsystem job header file:

```
#include <jde.h>
/************************************************************
 Report : R98SSUBE * ReportId : 8123244 * DSTRId : 380813 *
```

```
* Note: * Do not edit the following typedef
* To make modifications, use the Report Design Aid Tool to
 Generate a revised version.
*********************************************************/
#ifndef REPORT_DS_380813 #define REPORT_DS_380813
typedef struct tagDS_RI_380813 { char
ProgramId[11]; } DSRI380813, *LPDSRI380813;
#define IDERRProgramId_1         1L
#endif /* #define REPORT_DS_380813*/
#endif /* #define_R98SSUBE_H */
```

4. Add the header file to the business function using Microsoft Visual C++.
5. In the program to be called, call the business function using the appropriate event in Event Rules Design.

For example, you can create event rules that call a business function when the user clicks OK in the Oracle's JD Edwards EnterpriseOne Sales Order Entry (P4210) application. The business function uses the report's data structure and the subsystem's APIs to trigger the subsystem report. The API record retrieves the data structure and user information from the cache. If the server name is not passed, the API finds Object Map Record from the Data Source Master (F98611) table. If the record exists, it uses the record to send a JDENet Message to the server's subsystem kernel to add the record to the subsystem table on the server. However, if the user provides an override server name, the JDENet message is sent to that server's subsystem kernel instead. Each server is allowed to have only one subsystem kernel running.

Prerequisites

Before you begin defining subsystem jobs, ensure that you:

- Create a batch application object.
- Complete the design of the report template.
- Create a data structure, including the appropriate data items, for use in the business function.

Defining Reports as Subsystem Jobs

Open the report template in RDA.

1. From the File menu, select Report Data Structure.
2. On the Report Data Structure form, enter a data item alias on the QBE line to display the required data item.
3. From Dictionary Items, drag the required data item over to Structure Members and click OK.
4. From the File menu, select Report Properties.
5. On the Properties form, select the Advanced tab and select the Subsystem option.
6. Enter an appropriate value in the Wait Time (ms) field.

   The value in the Wait Time field is entered in milliseconds (1000 milliseconds equals one second). This represents the time that the subsystem job sleeps until checking the subsystem file for new records to process.
7. Click the Generate button to create a header file.

Adding Records to the Subsystem Table

Add a business function object in Oracle's JD Edwards EnterpriseOne Object Management Workbench.

1. On the Business Function Design form, select the Design Tools tab and click Start Business Function Design Aid.
2. Complete the function name and description and then attach the appropriate data structure.
3. In Microsoft Visual C++, open the business function include (.h) file and add the name of the header file that you generated.

   For example, the header file statement is emphasize in this business function include (.h) file:

```
/***************************************************************************
 Table Header Inclusions
***************************************************************************/

/**************************************************************************/
External Business Function Header Inclusions
***************************************************************************/
#include <R98SSUBE.h>
/***************************************************************************
 Global Definitions
***************************************************************************/

/***************************************************************************
* Structure Definitions
***************************************************************************/
* TYPEDEF for Data Structure
* Template Name:  Report Interconnect Data Structure
* Template ID:   D983059
* Generated:   Wed Oct 18 14:01:22 1995
```

4. In Microsoft Visual C++, open the business function source (.c) file and add lines to declare the variable of this data structure type and populate the members of the data structure.

   For example, the variable declaration for the report interconnect data structure is emphasize in this business function source (.c) file:

```
#include <jde.h>
/****************************************************************************
* Variable declarations
****************************************************************************/
HUSER hUser=NULL;
LPSTR szServer=NULL;
DSRI380813 dsRI;
/* Declare the variable of type REPORT INTERCONNECT DATA STRUCTURE */
BOOL bRet= FALSE;
JDEDB_RESULT rcode;
/****************************************************************************
Declare structures
****************************************************************************/

/****************************************************************************
Declare pointers
****************************************************************************/

/****************************************************************************
```

```
Check for NULL pointers
*****************************************************************************/
if ((lpBhvrCom == NULL) ||
  (lpVoid == NULL)  ||
  (lpDS == NULL))
```

5. In the business function source (.c) file, call the API to add the record to the subsystem table.

   For example, the API call to add the record to the subsystem table is emphasize in this business function source (.c) file:

   ```
   *****************************************************************************
   Main Processing
   *****************************************************************************/
   memset(&dsRI, 0, sizeof(DSRI380813));
   /* Populate the members of the Report Interconnect Data Structure */
    strcpy(dsRI.ProgramId,lpDS->szString01);
   /* Call Subsystem API to add the record to the Subsystem Table */
   /* Note : As Environment Name is set to NULL, this API will use OCM to find the#
    default
   Environment of this UBE */
   bRet=ubeReport_AddSubsystemRecord( hUser,/* User Handle */
      "R98SSUBE", /* Name of the subsystem */
      "XJDEOOO1", /* Name of the Subsystem Version*/
      NULL, /* Name of the override env - not used */
      szServer,  /* Name of the server */
    &dsRI);   /* Subsystem Connect DS */
   /**************************************************************************/
   ```

6. After you call the ubeReport_AddSubsystemRecord API to attach the record, build the business function.
7. Call the business function from the event rules process of the program to be called. This program can be either an interactive or batch application.

Note:

- *JD Edwards EnterpriseOne Tools APIs and Business Functions Guide* .
- *JD Edwards EnterpriseOne Tools Event Rules Guide* .

30 Creating Report Director Templates

Understanding Report Director Templates

The Director uses Report Director templates to aid you in designing reports. In addition to creating group, columnar, and tabular sections, the Director uses templates to help you create application reports, such as:

- Financial reporting
- Fixed assets
- Job cost

The Report Director templates guide you through the design process by presenting parameters that are relative to the specific application type. These Report Director templates, included with Oracle's JD Edwards EnterpriseOne, contain default criteria. When you select one of the templates in the Director, the system reads the template specifications, stored in JD Edwards EnterpriseOne tables, and presents the default criteria through the Director forms. Report Directors are similar to the Director that you use when creating columnar or group section report templates.

This table describes the template specifications that are stored in JD Edwards EnterpriseOne tables:

Table	Description
Report Director Templates (F91400)	Contains the default business view and processing option information.
Report Director Templates Sequence Items (F91410)	Contains information about the preferred data sequencing.
Report Director Templates Smart Field Activation (F91420)	Contains information about which smart fields to display.

Adding and Modifying Report Director Templates

This section provides an overview of Report Director template definitions and discusses how to create custom Report Director templates.

Understanding Report Director Template Definitions

You can modify JD Edwards EnterpriseOne Report Director templates and create custom templates through the Oracle's JD Edwards EnterpriseOne Report Director Templates (P91400) application. You can access the JD Edwards EnterpriseOne Report Director Templates application from both the Microsoft Windows client and the web client. When complete, the templates are available on the welcome form of the Director.

The Report Director template description appears in the Application Report drop-down list box on the welcome form of the Director. Custom Report Director template names should begin with *DT* and follow the recommended naming convention for JD Edwards EnterpriseOne objects.

One of the specifications that you can define for Report Director templates is the smart field template to be used. This is an optional specification. You can use Report Director templates without smart fields. However, you cannot use smart fields without Report Director templates.

The JD Edwards EnterpriseOne Report Director Templates application includes three tabs:

- Building Blocks

  Available for all Report Director templates.
- Properties

  Available only if the tabular detail section is selected for the report.
- Drill Down

  Available only if the tabular detail section is selected for the report.

For each Report Director template that you create, you can define these specifications:

- Which detail section type to use in the report.
- Which business view to use as the default.
- Which processing option template to use with the report.
- Which smart field template to make available for the selection layout.

  This is an optional specification.

  > **Note:** Smart fields associated with a Report Director template rely on the business view attached to the report template. If you select a different business view when designing the report, the associated smart fields might not function correctly. Before making any changes, ensure that you know which business view columns the smart fields require. This critical association is justification for defining the default business view in the Report Director template.

- The preferred section data sequencing and level breaks.

  Under the Default Sequence and Level Breaks heading, you can define the data sequencing to be used in the report. The first two data items that you select are defined as level break fields. These can be modified during the report design process.
- Which additional properties to include.

  Display financial or generic criteria. Available only if the tabular detail section is selected for the report.
- Define the appropriate Drill Down feature.

  Activate and define the Drill Down feature. Available only if the tabular detail section is selected for the report.

> **Note:**
> - *Creating Custom Smart Fields*.

Forms Used to Add Report Director Templates

Form Name	FormID	Navigation	Usage
Work With Report Director Templates	W91400A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Report Director Templates	Add, modify, and delete Report Director templates.
Report Director Templates Revisions	W91400B	Click Add on the Work With Report Director Templates form.	Enter the Report Director template name, description, section type, default business view, processing options, smart field template name (if appropriate), default data sequencing and level breaks, properties, and drill down information.

Creating Custom Report Director Templates

Access the Building Blocks tab of the Report Director Templates Revisions form.

Report Template/Description
Enter the name of the Report Director template and a meaningful description.

The system displays the Report Template/Description information from all three tabs.

Section Type
Enter the type of report section that the system should create when this Report Director template is selected.

Business View
Enter the business view that should be used by the report section. You can override this business view in the Director.

Processing Options
This field is optional. Enter the processing option template to be used by the report. You can override this processing option template in the Director.

Smart Field Template
This field is optional. Enter the smart field template to be used by the report. The smart field template is dependent on the business view defined.

Default Sequence and Level Breaks
Define the data sequencing and level breaks to be used by the report. These fields appear on the Data Sequencing Help form of the Director. The first two data items appear in Report Grouping and the subsequent data items appear in Report Detail. You can override these fields in the Director.

The system displays the Default Sequence and Level Breaks information from all three tabs.

Access the Properties tab of the Report Director Templates Revisions form.

Use Financial Description
Select this option to include the Use Financial Description option in the Director.

Provides the correct description of the object account. The system sets this option on the Additional Properties form of the Director.

Display Level of Detail
Select this option to include the Display Level of Detail option in the Director.

Determines the level of detail for summarizing the object account information based on the levels defined in the chart of accounts. The system sets this option on the Additional Properties form of the Director.

Display AAI Subtotal
Select this option to include the Display AAI Subtotal option in the Director.

Inserts subtotals based on the Financial Statements series of AAIs. The system sets this option on the Additional Properties form of the Director.

Display Adjust Sign
Select this option to include the Display Adjust Sign option in the Director.

Reverses the sign of amounts (debits and credits) in accounts for balance sheet or income statement reports. The system sets this option on the Additional Properties form of the Director.

Display Suppress Zero Rows
Select this option to include the Display Suppress Zero Rows option in the Director.

Suppresses the printing of a row in a tabular section using one of these options:

- No Zero Row Suppression.
- Suppress Zero Detail Rows Only.
- Suppress All Zero Rows.

Display Generic Criteria
Select this option to include generic criteria in the Director.

The Director displays the Section Data Selection form.

Display Financial Criteria
Select this option to include financial criteria in the Director.

The Director displays the Help with Section Data Selection form. This form enables you to select records based on balance sheet or income statement accounts.

> **Note:** The Display Generic Criteria and the Display Financial Criteria options are mutually exclusive.

Access the Drill Down tab of the Report Director Templates Revisions form.

Drill Down
Select to activate the drill feature in the report template. The application, form, and version must be defined to ensure that the Drill Down feature functions.

Drill Down App/Form/Vers

Select the name of the JD Edwards EnterpriseOne application, form, and version to launch to display the detail of the record selected in the report.

31 Creating Smart Fields

Understanding Smart Fields

Smart fields are data dictionary items with business functions attached. They are reusable objects that are created in the data dictionary and defined as glossary group K. The attached business function performs a specific task, such as a calculation, for the smart field.

As with all business functions, smart field business functions require a data structure to pass values. Smart field data structures include a named mapping that maps the source of each parameter included in the data structure. Named mappings are used in smart fields only.

Smart fields simplify the use of business functions because the parameters that need to be passed are held by the system. Instead of needing to know which business function to use and what parameters to pass, the user selects a smart field that inherently includes this information. Smart fields can be used for deriving column headings or populating values in a report section using Oracle's JD Edwards EnterpriseOne Report Design Aid (RDA).

For example, you can create a smart field to calculate sales by period. You can create the business function to add period 1, period 2, and period 3 to populate sales for the first quarter in a report. This calculation is performed by the business function for each row of data that is fetched by the report. Every time you use this smart field, it performs this calculation.

Without the quarterly smart field, you must write an event rule to accumulate the three periods to populate the quarterly sales column. To display a total for each quarter, you must write four event rules. Additionally, to display quarterly sales totals in multiple reports, you must duplicate these event rules in each report template.

Smart fields can be used in:

- Columnar sections
- Group sections
- Tabular sections

When you select a smart field for a detail section, you must define a column heading unless you use a smart field to populate the column heading. Typically, you define data selection for each smart field column. The exception is when the section data selection is the same as the data selection for each smart field column. In this instance, you can define data selection for the section. However, if you define data selection for each smart field column, you should still define data selection for the section for processing efficiency.

Before smart fields can be used in a report, you must add them to a Smart Field template and then add the Smart Field template to a Report Director template. You can add smart fields to an existing Smart Field template, if appropriate, or create a new Smart Field template. Smart Field templates organize smart fields that use the same fields for data selection. For example, Oracle's JD Edwards EnterpriseOne existing Smart Field templates are organized by Financial Reports, Fixed Assets, and 52-Period Accounting.

When you create a new Smart Field template, you must create a new Report Director template. In the Report Director template, you can define report processing options, business views, additional properties, and the Drill Down feature. Information that is included in the Report Director template guides the report developer through the process of creating a report template using the associated smart fields. The Smart Field template and Report Director template enable you to organize and present the smart fields in RDA.

Creating Custom Smart Fields

This section provides an overview of creating custom smart field components and discusses how to:

- Create data dictionary items.
- Create data structures.
- Define named mappings.
- Perform calculations using named event rules.
- Create data dictionary smart field items.

Understanding Smart Field Components

Smart fields are complex data items because they include business function logic. However, once the smart field is created, you do not have to recreate frequently used logic.

The basic components of smart fields are:

- Data dictionary items.

  A standard data item that is defined as glossary group D.
- Data structures.
- Named mappings.
- Business functions.

  Can use a C business function or a named event rule business function.
- Smart field data items.

  A data item that is defined as glossary group K.

Data Dictionary Items

The first component of most smart fields is a data dictionary item that serves as a user prompt for the report developer. The prompt indicates how the logic should be processed if more than one option is possible. For example, a quarterly sales smart field requires input from the report developer to indicate which quarter to calculate. At least four options are possible, one for each quarter, and each option results in a different outcome. However, not all smart fields require a user prompt. A smart field used to concatenate two fields does not require input as there are no decision points.

The Report Director (Director) displays a Smart Field Parameters form to prompt the report developer. The description of the data item prompt (for example, Quarter to Display) appears on this form. The glossary that you enter when you create the data item also appears on this form. The glossary explains the purpose of the prompt and assists the report developer in determining the appropriate value to enter in the prompt.

An example glossary:

- Enter 1 to print first quarter sales.
- Enter 2 to print second quarter sales.
- Enter 3 to print third quarter sales.
- Enter 4 to print fourth quarter sales.

Data dictionary items include a name, an alias, and a description. The data item name can be a maximum of 40 characters and cannot include spaces. The alias is a unique alphanumeric identifier. The software allows the alias to be a maximum of 10 characters. However, not all databases support 10 characters, so it is recommended that you limit the alias to eight characters. The description can be a maximum of 40 characters, including spaces. The description is often the same as the name, with spaces.

See *JD Edwards EnterpriseOne Tools Data Dictionary Guide* .

Data Structures

The second component that is required for a smart field is a data structure. A data structure lists the parameters that pass values between a report and the database tables. You must include all of the data items that are required to complete the function of the smart field in the data structure. For example, a quarterly amount smart field requires 12 periods for calculating each month. Another field is required for the return value, which is the result of the smart field calculation for each quarter.

All data items that are added to the data structure must reside in the same business view. If you find that you need to add data items that are not included in a single business view, you must create a custom business view that includes all of the required data items. Adding data items that reside in different business views results in a nonfunctional smart field.

You can also define the direction of the flow of data in a smart field data structure. This definition eliminates the need for the report developer to define the flow in RDA. The data flow definition is another component that simplifies the use of business functions for the report developer.

See *"Creating Data Structures," "Creating Business Function Data Structures" in the JD Edwards EnterpriseOne Tools Data Structure Design Guide* .

Named Mappings

Named mappings are a component of the data structure that is used only for smart fields. Named mappings define each of the data items that is included in the data structure. They can also hold default values for the business function so that the values do not need to be manually defined in RDA.

Named mappings are also used to map the source for each parameter (or data item) of the data structure. For example, source values are determined for the prompts, tables, and return values. This table details the sources from which data structure data items originate:

Source	Description
Literal	Assigns a specific value to the data item. For example, if the calculation needed to use a tax rate, you would enter the tax rate in the value field.
Prompt	Indicates that the data item is the user prompt. For example, the Quarter to Display data item is the prompt. In RDA, the report developer enters the quarter to calculate, and the result appears in the smart field column of the report.
Table	Indicates the data items that originate from a table. Browse to locate the table name, and then associate the data item in the data structure with the corresponding data item in the table.
Data Dictionary Item	Passes values from a processing option into the data structure. Values in smart fields can be identified with a business function without requiring input. For example, the desired fiscal year or period might already be specified in a processing option that can be passed into the data structure. If this is the case,

Source	Description
	the data item must be defined as both a data dictionary item and a processing option in the named mapping.
System Value	Associates the system value as the origin of the data item and browses for the appropriate system value. System values, such as system date, are used throughout the system. System values are fetched from the Table of Variables (F98VAR) table.

Named mappings begin with the letter *M* and use the same name as the data structure. You can add multiple named mappings to a single data structure. If you have multiple named mappings, append a sequential letter of the alphabet to the named mapping name. For example, the first named mapping for data structure D550101 would be named M550101A.

Business Functions

Business functions define the logic for the smart field. You can use either C business functions or named event rule business functions. C business functions are written in C language, while named event rule business functions are written in scripting language using the JD Edwards EnterpriseOne toolset. Named event rule business functions are also referred to as named event rules (NERs).

The advantage of using business functions is that they are reusable. The code is written once and can be used in multiple events and reports. For example, without the business function for Quarterly Amount, you would write four event rules, adding together the appropriate periods for each of the four quarters. In contrast, if you use an NER you can write the criteria once and reuse it for each of the four columns that displays a quarterly amount. You can also use this same NER in other reports that require quarterly sale figures.

The event rules scripting language used for creating NERs is platform independent. NERs are stored in a database as a JD Edwards EnterpriseOne object. When you build NERs, the system generates C code and creates .c and .h files.

A NER is a business function object. When you create a business function, you have the choice of which source language to use. The source language options are *C* and *NER.* Your selection determines which tool the system presents.

As with any business function, you need to associate a data structure with an NER.

You create the logic for NERs using the Named Event Rules Design form. This form is similar to the Event Rules Design form used in RDA. In the Named Event Rules Design form, you can create event rule logic for each valid user prompt value. It is good practice to include a default statement. For example, the logic for quarterly amount might state that if any value other than 1, 2, 3, or 4 is entered into the user prompt, the first quarter is calculated by default.

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

Smart Field Data Items

The last component that is required for a smart field is a data dictionary item that is defined as a smart field. Smart field data dictionary items differ from regular data dictionary items in two ways:

- Glossary group.

  Regular data dictionary items are classified as glossary group D; smart field data items are classified as glossary group K. The K classification enables the Smart Field Criteria form.

- Smart Field Criteria form.

  The Smart Field Criteria form defines the business functions and named mappings that are associated with the smart field and the event on which the logic is attached.

  If you create smart fields for use in tabular sections, you might select the Column Inclusion event. If you create smart fields for use in columnar or group sections, you might select the Do Variable event.

> **Note:**
>
> - *JD Edwards EnterpriseOne Tools Table Conversion Guide* .
> - *JD Edwards EnterpriseOne Tools Data Structure Design Guide* .

Creating Data Dictionary Items

From the Tools menu in JD Edwards Solution Explorer, select Object Management Workbench (OMW).

1. Select the project in which you want to add a new data dictionary item and click Add.
2. On the Add EnterpriseOne Object to the Project form, select Data Item and click OK.
3. On the Data Dictionary Item Type form, click No.

   This is a regular data dictionary item, not a glossary data item.
4. On the Data Item Specifications form, enter the name of the data item in the Data Item field.
5. In the Alias field, enter a unique alphanumeric identifier.
6. In the Glossary Group field, enter **D** to indicate a primary data element.
7. In the Description field, enter a brief description of the data item.
8. In the Product Code field, enter a product code.

   Codes in the 55–59 range are reserved for clients.
9. In the Product Code Reporting field, enter a product code that represents the system associated with this data item.
10. In the Data Type field, use the visual assist to select a value that indicates the type of data to be entered (for example, character, string, numeric).
11. In the Size field, enter the field size of the data item.

    This field is enabled based on the data type you selected.
12. In the File Decimals field, enter the number of stored positions to the right of the decimal place.

    This field is enabled only if the data type is defined as numeric.
13. In the Class field, use the visual assist to select a class that defines the attributes and characteristics of the data item (for example, CURRENCY).

    This field is only informational; it is not required.
14. In the Display Decimals field, enter the number of decimals that you want to appear.

    This field is enabled only if the data type is defined as numeric.
15. In the Control Type field, enter the type of control that is associated with the data item (for example, check box, push button, generic edit, and so on).

    The generic edit option is typically used when creating data items specifically for batch applications.

16. Leave the Item Occurrences field cleared.

    The value entered in this field causes a number of identical data items to be created based on the template defined by the active data item properties. For example, if data item RDATEST is added with Item Occurrences set to **2,** the system creates two copies of the data item, RDATEST1 and RDATEST2.
17. In the Row Description field, enter the data item description as it should appear in group sections.
18. In the Column Title fields, enter the data item description as it should appear in columnar sections.

    Split lengthy descriptions into two lines. Enter short descriptions in the top line.
19. Select the Item Glossary tab, enter a description of all of the valid values for this prompt, and click OK.

    The item glossary appears on the Smart Field Parameters form in the Director.

    For example, for a quarterly data item prompt you might enter:

    Enter 1 to display totals for the first quarter.

    Enter 2 to display totals for the second quarter.

    Enter 3 to display totals for the third quarter.

    Enter 4 to display totals for the fourth quarter.

Creating Data Structures

Access the appropriate project in OMW.

1. Add a data structure to the same project in which you added the data item smart field component.
2. On the Add Object form, create a new regular data structure, name it according to the recommended naming conventions, and click OK.

   See *"Understanding JD Edwards EnterpriseOne Naming Conventions" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .
3. On the Data Structure Design form, select the Design Tools tab and click Data Structure Design.
4. On the Data Structure form, under Dictionary Items, enter the alias of the data item prompt on the QBE line.
5. Double-click the data item prompt to include it in Structure Members.
6. Double-click in the Required field of the detail area to define the data item prompt as Required.

   This field displays a check mark for the prompt data item.
7. Under Dictionary Items, search for all of the required data items using the QBE line.
8. Double-click the required data items individually to include them in Structure Members.
9. Under Dictionary Items, search for a data item to serve as a return value using the QBE line.

   Use a data item that is formatted the same as you want the smart field to be formatted.
10. Double-click the return value data item to include it in Structure Members.

    You can rename the return value data item in the Structure Member Name field.
11. In the Input/Output field for each data item, set up the arrows to reflect the flow of the data and click OK.

    All arrows for prompts and data items should point to the right because these fields fetch data from the database. The arrow for the return value should point to the left because the data is returned to this field in the report section.

Defining Named Mappings

In OMW, select the smart field data structure, click the Design button, and select the Design Tools tab.

1. Click Named Mapping.
2. On the Named Mapping form, click Add.

   The detail area includes all data items included in the data structure.
3. On the Add Argument List form, enter the name of the named mapping using the recommended naming conventions.

   See *"Understanding JD Edwards EnterpriseOne Naming Conventions" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .
4. On the Named Mapping form, click the prompt data item and select the Prompt option under Origin Types.

   The prompt is already set as a required data item from the data structure definition.
5. Click one of the data items to be fetched from a table and select the Table option under Origin Types.
6. Click the Browse button to access the Select a Table form.
7. Use the QBE (query by example) line to locate the table where the data item resides, and click Next.
8. On the Select a Column form, select the field that matches the data structure data item and click Finish.
9. On the Named Mapping form, define each of the data items that is fetched from a table using the Required option at the bottom of the form.
10. Define the return value data item using the Return Value option at the bottom of the form.

    > **Note:** When you select the Return Value option, the system automatically selects the Required option.

Performing Calculations Using Named Event Rules

Access the appropriate project in OMW.

1. Add a business function to the same project in which you added the other smart field components.
2. On the Add Object form, create a new business function, naming it according to the recommended naming conventions.
3. Under Source Language, select NER and then click OK.
4. On the Design Tools tab of the Business Function Design form, click Start Business Function Design Aid.
5. On the Business Function Design form, click the visual assist in the Parent DLL field and select the DLL that your company uses to create custom business functions.
6. 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.
7. Enter a description in the Description field.

   This description can be the same as the name, however, you can use spaces in the description.
8. Tab through the remaining fields.

   The system populates the F3 Code field with a 3 (Minor Business Rule).
9. Modify the F3 Code if appropriate.
10. Click the row header to highlight the row of data that you just entered, and select Parameters from the Row menu.

11. Use the QBE line to locate the associated data structure, click the data structure and then click Select.
12. 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.
13. From the Form menu, select Edit.
14. On the Named Event Rules Design form, create the appropriate logic using If/While statements, assignments, business functions, and system functions.

    For example, for the quarterly amount smart field that calculates sales for each quarter, this logic is appropriate:

    ```
    If <user prompt> is equal to 2
    <return value> = Sales for Period 4 + Sales for Period 5 + Sales for Period 6
    Else
    If <user prompt> is equal to 3
    <return value> = Sales for Period 7 + Sales for Period 8 + Sales for Period 9
    Else
    If <user prompt> is equal to 4
    <return value> = Sales for Period 10 + Sales for Period 11 + Sales for Period 12
    Else
    <return value> = Sales for Period 1 + Sales for Period 2 + Sales for Period 3
    End If
    End If
    End If
    ```

    > **Note:** The first quarter sales is the default result. Therefore, if the user enters a value other than 2, 3, or 4, the business function calculates sales for the first quarter.

15. When the logic is complete, save the event rules and click OK.
16. On the Design Tools tab, click Build Business Function.

Creating Data Dictionary Smart Field Items

Access the appropriate project in OMW.

1. Add a data item to the same project in which you added the other smart field components.
2. On the Data Dictionary Item Type form, click No.

   This in not a glossary data item so you want to answer no to the message form.
3. On the Data Item Specifications form, enter the name of the data item in the Data Item field.
4. In the Alias field, enter a unique alphanumeric identifier.
5. In the Glossary Group field, enter **K** to indicate that the data item is a smart field.
6. In the Description field, enter a brief description of the smart field data item.
7. In the Product Code field, enter a product code in the 55–59 range, codes reserved for clients.
8. In the Product Code Reporting field, enter a product code that represents the system associated with this smart field data item.
9. In the Data Type field, use the visual assist to select a value that indicates the type of data to be entered (for example, character, string, numeric).
10. In the Size field, enter the field size of the smart field data item.
11. In the File Decimals field, enter the number of stored positions to the right of the decimal place.
12. In the Class field, use the visual assist to select a class that defines the attributes and characteristics of the smart field data item (for example, CURRENCY).
13. In the Display Decimals field, enter the number of decimals to appear.
14. In the Control Type field, enter the type of control that is associated with the smart field data item (for example, check box, push button, generic edit, and so on).

The generic edit option is typically used when creating data items specifically for batch applications.

15. Leave the Item Occurrences field cleared.
16. In the Row Description field, enter the smart field data item description as it should appear in group sections.
17. In the Column Title fields, enter the smart field data item description as it should appear in columnar sections.
18. Select the Item Glossary tab and enter information that describes all of the valid results.

    For example, for a quarterly smart field, you might enter:

    - **1 returns a total value for January, February, and March.**
    - **2 returns a total value for April, May, and June.**
    - **3 returns a total value for July, August, and September.**
    - **4 returns a total value for October, November, and December.**
19. From the Form menu, select Smart Field to launch the Smart Field Criteria form.
20. In the Business Function field, use the visual assist to locate the appropriate business function, click the business function, and then click Select.
21. In the Event Name field, use the visual assist to select the event from which the smart field business function will be called.
22. In the Named Mapping field, enter the name of the named mapping that you defined for the associated data structure and click OK.

> **Note:**
>
> - *JD Edwards EnterpriseOne Tools Data Dictionary Guide* .

Creating Custom Smart Field Templates and Reports

This section provides an overview of creating custom smart field template and discusses how to:

- Create Smart Field templates.
- Create Report Director templates.
- Design reports using custom smart fields.

Understanding Custom Smart Field Template Creation

After you create custom smart fields, you must add them to a Smart Field template and define the appropriate data selection. You must then add the smart field template to a Report Director template. The Report Director template makes the smart fields available for use in report templates.

Smart Field Templates

JD Edwards EnterpriseOne includes predefined Smart Field templates. Smart Field templates group smart fields that use the same data selection criteria so that you can include them collectively in a Report Director template. For example, the S09001 - Financial Reporting smart field template contains all of the smart fields for financial reporting. Through the Oracle's JD Edwards EnterpriseOne Smart Field Templates (P91420) application, you can add new smart field templates or modify existing smart field templates to meet your reporting needs. You can access the JD Edwards EnterpriseOne Smart Field Template application from both the Microsoft Windows client and the web client.

You must add smart fields to a smart field template before they can be used in reports. Then you must attach the smart field template to a Report Director template. When you use the Report Director template to create a report, the smart fields in the attached smart field template appear in the Available Smart Fields list on the Select Columns form of the Director. From this form, you can select any of the available smart fields to include in the report.

When you select a smart field to include in the section layout, you are typically prompted for data selection values. Data selection fields are defined within the smart field template. These fields appear on the Smart Field Data Selection form of the Director.

Custom smart field template names should begin with *SFT.* The next two characters should be one of the product codes that is reserved for clients (55–59), and the next characters should reflect the system that uses the smart field template, as recommended for JD Edwards EnterpriseOne objects.

When you define a smart field template, you can select multiple smart fields to include in the template as long as they all use the same data selection. When defining data selection, you must select from fields that reside in the business view that is associated with the smart fields. Only five data items can be included in the smart field data selection criteria.

You can define a data selection field as requiring a range of values, but the field must be counted as two items. A range of values takes up two lines on the Smart Field Data Selection form in RDA. For example, if you include business unit as a data selection item, you might want the report developer to be able to include a range of values. This option presents two fields on the Smart Field Data Selection form in the Director, one for the beginning business unit value and one for the ending business unit value.

Report Director Templates

Report Director templates define a set of parameters to guide the report developer through the creation of a batch application. Report Director templates are typically created to use smart fields. However, you can use Report Director templates to define parameters for the development of batch applications without the use of smart fields.

When you create a Report Director template, you define the appropriate business view to attach to the report section. The business view that you select must include all of the data items included in the smart field data structure. It is this business view that is used when the report developer selects the option I'll use the predefined business view from the Director.

Custom Report Director templates should begin with *DT.* The next two characters should be one of the product codes that is reserved for clients (55–59), and the next characters should reflect the system that uses the Report Director template.

The Report Director template description appears in the Application Report drop-down list box on the Director Welcome form and on the Business View Selection Options form when you create the report in RDA.

Report Templates

As the final step in the smart field creation process, you should test the smart field. It is helpful to review the Director and see how all of the information you defined is presented to the report developer. This review helps you understand the origin of each option presented in the Director and how your selections in defining the objects affect what the report developer sees.

Smart Field Logic

You can review the event rules that are created by the smart fields to understand how the smart field components are implemented in the report. In the RDA workspace, click the variable portion of a smart field column. From the Edit menu, select Event Rules. Select the event that you defined on the Smart Field Criteria form when you created the smart field data item.

The description of the business function that you created for the smart field appears on the event. Double-click the business function to review the data structure parameters. The business function information appears in the upper-left corner of the Business Functions form and includes the business function name, description, and the name of the attached data structure. The data structure displays the parameters as you defined them in the named mapping.

Notice that the data structure includes the directional arrows you defined in the data structure. The directional arrows for all data items except the return value data item point to the right. This is because each of these fields is fetched from a source, such as the database or a user prompt. The system then processes the logic using these fields and returns the result to the return value field to appear on the report. The directional arrow in this field points to the left. Without the use of this smart field, the report developer would have to know which business function to use and how to pass each of these values.

For example, if you select the first quarter variable that was created using the quarterly sales smart field, you can see in the data structure the **1** that you entered on the Smart Field Parameters form when you created the report.

See *"Understanding JD Edwards EnterpriseOne Naming Conventions" in the JD Edwards EnterpriseOne Tools Development Guidelines for Application Design Guide* .

Forms Used to Create Smart Field Templates and Reports

Form Name	FormID	Navigation	Usage
Work With Smart Field Templates	W91420A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Smart Field Templates	Add, modify, and delete smart field templates.
Smart Field Template Revisions	W91420G	Click Add on the Work With Smart Field Templates form.	Enter smart field template name, description, and smart field data item.
Smart Field Template Criteria Revisions	W91420B	Click OK on the Smart Field Template Revisions form.	Enter fields to be used for data selection.
Work With Report Director Templates	W91400A	EnterpriseOne Life Cycle Tools, Report Management (GH9111), Advanced Report Setup, Report Director Templates	Add, modify, and delete Report Director templates.
Report Director Templates Revisions	W91400B	Click Add on the Work With Report Director Templates form.	Enter the Report Director template name, description, section type, default business view, processing options, smart field template name (if appropriate), default data sequencing and level breaks, properties, and drill down information.

Creating Smart Field Templates

The smart field template includes two parts, the Smart Field Template Revisions form and the Smart Field Template Criteria Revisions form.

Smart Field Template Revisions

Access the Smart Field Template Revisions form.

Smart Field Template
Enter the name of the Smart Field template.

Description
Enter a meaningful description of the type of smart fields that are included in the Smart Field template.

Data Item and Description
Enter the data item alias of all smart fields to be included in the Smart Field template.

Description
Describes the data item based on the alias entered. The system populates this field.

Smart Field Template Criteria Revisions

Access the Smart Field Template Criteria Revisions form.

Smart Field Template
Displays the name of the Smart Field template as entered on the Smart Field Template Revisions form.

Description
Displays the description of the type of smart fields that are included in the Smart Field template as entered on the Smart Field Template Revisions form.

Data Item
Enter the alias of the data items to be used as data selection in the report sections created using the Smart Field template.

Description
Describes the data items to be used as data selection in the report sections created using the Smart Field template. The system populates the description field based on the alias that you enter.

Display Seq
Indicates the sequence in which the data selection fields appear on the Smart Field Data Selection form of the Director. The system populates this field when you tab out of it.

Range Values
Indicates whether a field requires a range of values or a single value. The system populates this field with a default value of **0,** but you can override this value. For example, if Units-Order Transaction Quantity requires a range of values, enter **1** in the field.

Creating Report Director Templates

Access the Report Director Templates Revisions form.

Report Template/Description
Enter the name of the Report Director template and a meaningful description.

The system displays the Report Template/Description information from all three tabs.

Section Type
Select a user-defined code (UDC) that represents the type of report section that the system creates when this Report Director template is selected.

Business View
Enter the business view to be used by the report. You can override this business view in the Director.

Processing Options
This field is optional. Enter the processing option template to be used by the report. You can override this processing option template in the Director.

Smart Field Template
Enter the smart field template to be used by the report. The smart field template is dependent on the business view defined.

Default Sequence and Level Breaks
The system displays the Default Sequence and Level Breaks information from all three tabs.

Data Item
Enter the alias of the data items to be used as data sequencing and level breaks in the report sections created using the Report Director template. The first two data items appear in Report Grouping, and the subsequent data items appear in Report Detail on the Data Sequencing Help form of the Director. You can override these fields in the Director.

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.

> **Note:**
> - *Creating Report Director Templates*.

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.

32 Defining 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.

Note:

- *"Implementing Error Messages" in the JD Edwards EnterpriseOne Tools Development Standards for Business Function Programming Guide* .

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, 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.

> **Note:**
>
> - *"Implementing Error Messages", "Using Text Substitution to Display Specific Error Messages" in the JD Edwards EnterpriseOne Tools Development Standards for Business Function Programming Guide* .

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:

```
jdeStrcpy((JCHAR*)szDsTmpl,)const JCHAR*)(_J("DELM0002)),DIM(szDsTmpl)-1);
jdeStrcpy(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.szEditransactnumber`** is located in the text substitution data structure and **`pDS->szEditransactnumber`** 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->szEdiuserid),
 DIM(dsFormData.EDUS));
jdeStrncpy(dsFormData.EDBT,(const JCHAR *)(lpDS->szEdibatchnumber),
 DIM(dsFormData.EDBT));
jdeStrncpy(dsFormData.EDTN,(const JCHAR *)(lpDS->szEditransactnumber),
 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 *)lpDSwork->lpBlob->szForm,(const JCHAR *)(szForm),
      DIM (lpDSwork->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,
        &lpDSwork->lpBlob->lpFIDSMPL);
   if(lpDSwork->lpBlob->lpFIDSMPL !=(LPDSTMPL)NULL)
   {
    lpDSwork->lpBlob->lpFITEXT=(PJSTR)AllocBuildStrFromDstmplName
                 (LPDSTMPL)
    lpDSwork->lpBlob->lpFIDSMPL,(JCHAR*)lpASVRdtl->szFITemplateName,
           (LPVOID)&dsFormData);
    jdeStrncpy(lpDSwork->lpBlob->szDLLName,(const JCHAR *)(szDLLName),
         DIM(lpDSwork->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 ItemsAlias 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.

33 Retrieving EnterpriseOne Report Runtime System Information using an API

Understanding EnterpriseOne Report Runtime System Information Using an API

Effective with JD Edwards EnterpriseOne Tools Release 9.2.7, an API and a related system function are provided to retrieve useful information of batch jobs from the virtual job batch queue. Examples of the information that can be retrieved by the API include the batch job number, hostname (virtual and real hostname), queue, and the report name and version. In turn, a related system function can pass this information to JD Edwards EnterpriseOne internal applications.

Business Function GetRuntimeInfo

The Business function GetRuntimeInfo (B986110R) allows access from the report ER to information related to the batch job processing environment at runtime. The data available includes:

1. Report Name
2. Version Name
3. User Name (short)
4. EnterpriseOne Environment Name
5. Time stamp when the spec load completed and Execution begins, in UTC (no timezone applied)
6. Job Queue – This value is blank on a Dev Client.
7. Execution Host Name – This is either the Virtual Host name of a VBQ configuration, or the Actual Hostname. It is always “WinClient” on a Dev Client.
8. Actual Host Name – This is the actual Enterprise Server host name.
9. Job Number (as a math numeric) – The current job number. Always 0 on dev clients.
10. Parent Job Number (as a math numeric) – This is the value is 0 if called in top-level UBE, otherwise this is the job number of UBE which called this function. Always 0 on dev clients.
11. Oldest Parent Job Number (as a math numeric) – This is the value is 0 if called in top-level UBE, otherwise this is the job number of topmost UBE of synchronous job chain. Always 0 on dev clients.
12. Job Number (as a string) – The current job number. Always “0” on dev clients.
13. Parent Job Number (as a string) – This is the value is “0” if called in top-level UBE, otherwise this is the job number of UBE which called this function. Always “0” on dev clients.
14. Oldest Parent Job Number (as a string) – This is the value is “0” if called in top-level UBE, otherwise this is the job number of topmost UBE of synchronous job chain. Always “0” on dev clients.
15. Printer Name
16. PDF File Name – This name is assigned only at completion of End Report Event. BI Publisher reports do not have a valid PDF name.
17. CSV File Name – This name is assigned only at completion of End Report Event, and is only valid for reports that have CSV output activated.

18. OSA filename – This name is assigned only at completion of End Report Event, and is only valid for reports that have an OSA output stream activated.

For reports running on a Development Client or Deployment Server, the server name will always be "WinClient". Further, on these machines the job numbers will always be 0. This is because jobs that run locally on these machines are not tracked using a job number in the WSJ (P986110B) application.

The files the filenames refer to are only guaranteed to be available at the End Report event. After this point, this information may be stored in the database and removed from disk according to the setting in the P98617 Output location application.

A typical event to put the ER to call the BSFN would be on the InitSection event of the main driver section of the report passing in section-level or report-level variable to be populated by the function.

BSFN Data Structure D986110R

The Data structure shows what Data Dictionary types to use for your ER variables that will be passed to the function. Reference also the B986110R.h include file, which also has a C data structure for use with the jdeCallObject API and BSFN GetRuntimeInfo in B986110R.c. The values in this structure are listed in the following table.

Structure Member Name	Data Item	Required?	Input/Output	Alias
szReportName	NameObject	No	Output	OBNM
szVersion	Version	no	Output	VERS
szUserId	UserId	no	Output	USER
szEnvironmentName	EnvironmentName	no	Output	ENHV
utExecutionStarted	UTIME	no	Output	UTIME
szJobQueue	JobQueue	no	Output	JOBQUE
szExecutionHost	ExecutionHostName	no	Output	EXEHOST
szActualHost	ExecutionHostName	no	Output	EXEHOST
mnEOneJobnum	ServerJobNumber	no	Output	JOBNBR
mnEOneParentJobnum	ServerJobNumber	no	Output	JOBNBR
mnEOneOldestParentJobnum	ServerJobNumber	no	Output	JOBNBR
szEOneJobNum	JobControlFutureUse1	no	Output	JOBFU1
szEOneParentJobNum	JobControlFutureUse1	no	Output	JOBFU1
szEOneOldestParentJobNum	JobControlFutureUse1	no	Output	JOBFU1
szPrinterName	PhysicalDevice	no	Output	PHYD
szPDFFilename	InitFileName	no	Output	FBINIFN

Structure Member Name	Data Item	Required?	Input/Output	Alias
szCSVFilename	InitFileName	no	Output	FBINIFN
szOSAFilename	InitFileName	no	Output	FBINIFN

System API

The system-level API named **ubeReport_GetRuntimeInfo()** is available as of JD Edwards EnterpriseOne Tools Release 9.2.7.0. This API is a tools-layer C function that is used by the business function GetRuntimeInfo to retrieve the information from the EnterpriseOne system at report runtime.

The header file `jdekdfn.h` in `system\include` contains an enum of the types of data that can be retrieved in the type UBE_INFO_TYPE. Each item type in this header is commented with usage notes. For detailed examples of use, refer to the GetRuntimeInfo() business function. It is important to know that this function returns a void pointer to memory allocated by the function containing the data requested, and that this pointer must be freed with a call to jdeFree() after use.

The following is a listing of the ENUM definitions for the **ubeReport_GetRuntimeInfo** BSFN API.

```
UBEGet_EOneJobnum_JCHAR = 1, // DD:JOBFU1 current jobnum (as string)
UBEGet_EOneJobnum_MATHNUM = 2, // DD:JOBNBR current jobnum
UBEGet_ExeHost_JCHAR = 3, // DD:EXEHOST current exehost may match actual host, or be a VBQ servername
UBEGet_ActualHost_JCHAR = 4, // DD:EXEHOST current actual enterprise sever hostname
UBEGet_QueueName_JCHAR = 5, // DD:JOBQUE queue name current job is running in
UBEGet_ReportName_JCHAR = 6, // DD:OBNM report name
UBEGet_VersionName_JCHAR = 7, // DD:VERS version name
UBEGet_ShortUserName_JCHAR = 8, // DD:USER short user name
UBEGet_EnvironmentName_JCHAR = 9, // DD:ENHV environment name
UBEGet_TimeExecutionStartUTC_UTIME = 10, // DD:UTIME time of spec load completed and Execution begins, in
 UTC (no timezone applied)
UBEGet_EOneParentJobnum_JCHAR = 11, // DD:JOBFU1 0 if called in top-level UBE, otherwise jobnum of UBE which
 called this one (as string)
UBEGet_EOneParentJobnum_MATHNUM = 12, // DD:JOBNBR 0 if called in top-level UBE, otherwise jobnum of UBE
 which called this one
UBEGet_EOneOldestParentJobnum_JCHAR = 13, // DD:JOBFU1 0 if called in top-level UBE, otherwise jobnum of
 topmost UBE of synchronous job chain (as string)
UBEGet_EOneOldestParentJobnum_MATHNUM = 14, // DD:JOBNBR 0 if called in top-level UBE, otherwise jobnum of
 topmost UBE of synchronous job chain
UBEGet_PrinterName_JCHAR = 15, // DD:PHYD Current printer name
UBEGet_PDFFilename_JCHAR = 16, // DD:FINAME Current PDF filename, if any (BIPub reports don't have a valid
 one at engine runtime) - only complete at End Report Event.
UBEGet_CSVFilename_JCHAR = 17, // DD:FINAME Current CSV filename, if any - only complete at End Report
 Event.
UBEGet_OSAFilename_JCHAR = 18 // DD:FINAME Current OSA filename, if any - only complete at End Report Event.
```

34 Glossary

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.

edit rule

A method used for formatting and validating user entries against a predefined rule or set of rules.

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.

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.

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.

named event rule (NER)

Encapsulated, reusable business logic created using event rules, rather that C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

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.

Index