
Oracle Developer: Build Reports

Volume 1 • Student Guide

43212GC10

Production 1.0

April 1999

M08692

ORACLE®

Authors

Christian Bauwens
Ursula Hovy

Technical Reviewers

Gunnar Boehrs
Jackie Bruce
Ruth Delaney
Laurent Dereac
Astrid Dietrich
Brian Fry
Mary Lee
Stella Misiulis
Paul Narth
Sergiy Pechersky
Raza Siddiqui
John Walsh

Publisher

Sherry Polm

Copyright © Oracle Corporation, 1999. All rights reserved.

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c) (1) (ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Box SB-6, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

All references to Oracle and Oracle products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only and may be trademarks of their respective owners.

Preface

- Profile xv
- Related Publications xvi
- Typographic Conventions xvii

Introduction

- Overview I-3
- Course Objectives I-5
- Course Content I-7

Lesson 1: Introduction to Oracle Reports

- Overview 1-3
- What Is Oracle Developer? 1-5
- Introducing the Oracle Developer Release 6.0 Components 1-9
- Report Builder Components 1-13
- Getting Started in the Oracle Reports Interface 1-17
- Navigating Around the Oracle Reports Main Menu 1-19
- Customizing Your Oracle Reports Session 1-21
- Oracle Reports Environment Variables 1-25
- Oracle Reports Help Options 1-27
- Introducing the Course Application 1-29
- Summary 1-31
- Practice Session Overview: Lesson 1 1-33
- Practice Session: Lesson 1 1-34

Lesson 2: Designing and Running Reports

- Overview 2-3
- Designing Reports 2-5
- Design Considerations 2-7
- Retrieving and Sharing Data 2-15
- Running a Report 2-17
- Viewing the Report Output 2-19
- Summary 2-29
- Practice Session Overview: Lesson 2 2-31
- Practice Session: Lesson 2 2-32

Lesson 3: Report Builder Concepts

- Overview 3-3
- Report Executables 3-5
- Report Builder Components 3-9
- Object Categories 3-13
- Summary 3-23
- Practice Session Overview: Lesson 3 3-25
- Practice Session: Lesson 3 3-26

Lesson 4: Creating Reports Using the Report Wizard

- Overview 4-3
- Invoking Report Builder and the Report Wizard 4-5
- Creating a Tabular Report Using the Wizard 4-7
- Creating Other Report Styles 4-23
- Summary 4-33
- Practice Session Overview: Lesson 4 4-35
- Practice Session: Lesson 4 4-36

Lesson 5: Enhancing Reports in the Live Previewer

- Overview 5-3
- What Is the Live Previewer? 5-5
- Modifying a Report in the Live Previewer 5-7
- Summary 5-22
- Practice Session Overview: Lesson 5 5-24
- Practice Session: Lesson 5 5-25

Lesson 6: Managing Report Templates

- Overview 6-3
- Using the Report Templates 6-5
- Modifying a Template 6-7
- Predefining Your Own Templates 6-13
- Summary 6-17
- Practice Session Overview: Lesson 6 6-19
- Practice Session: Lesson 6 6-20

Lesson 7: Report Storage Methods

- Overview 7-3
- Storage Methods and Requirements 7-5
- Documenting Your Reports 7-9
- File Portability 7-11
- Converting Report Definitions 7-15
- Migrating Reports from Previous Releases 7-17
- Summary 7-19
- Practice Session Overview: Lesson 7 7-21
- Practice Session: Lesson 7 7-22

Lesson 8: Enhancing Reports Using the Data Model: Creating Queries and Groups

- Overview 8-3
- Describing the Report Data Model 8-5
- Modifying Properties of a Query 8-7
- Changing the Group Structure 8-13
- The Group Hierarchy 8-15
- Ordering Data in a Group Hierarchy 8-17
- Filtering Data in a Group 8-21
- Adding New Data 8-23
- 8-27
- Oracle8 Objects in Report Builder 8-30
- Using Data Links to Structure Data 8-36
- Efficiency Issues: Multiple- or Single-Query Hierarchy 8-42
- Summary 8-44
- Practice Session Overview: Lesson 8 8-46
- Practice Session: Lesson 8 8-47

Lesson 9: Enhancing Reports Using the Data Model: Creating Columns

- Overview 9-3
- The Data Model Column Types 9-5
- Maintaining Database Columns 9-7
- Using a Database Column to Output Contents of a File 9-9

- Creating a Column 9-11
- Creating Summary Columns 9-13
- Example Report Displaying Subtotals 9-15
- Example Report Displaying Percentages 9-17
- Creating Formula Columns 9-21
- Creating Placeholder Columns 9-23
- Example Report to Populate a Placeholder 9-25
- Summary 9-27
- Practice Session Overview: Lesson 9 9-29
- Practice Session: Lesson 9 9-30

Lesson 10: Enhancing Reports Using the Layout Model

- Overview 10-3
- Viewing the Layout Model Regions 10-5
- Viewing the Report Sections in the Layout Model 10-7
- Designing Multipanel Reports 10-9
- The Layout Model Layers 10-13
- Creating Layout Objects 10-19
- The Layout Model Tools 10-21
- Creating Additional Layouts in a Report 10-23
- Creating Variable Length Lines 10-25
- Displaying a Multimedia File from a Button 10-27
- Creating Explicit Anchors 10-29
- Summary 10-33
- Practice Session Overview: Lesson 10 10-35
- Practice Session: Lesson 10 10-36

Lesson 11: Modifying Properties

- Overview 11-3
- Modifying Layout Object Properties 11-5
- Displaying File Contents with a Link File 11-7
- Modifying Common Layout Properties 11-13
- Sizing Objects 11-15
- Layout Object Relationships 11-17

Modifying Properties of a Field 11-29
 Modifying Properties of a Repeating Frame 11-37
 Summary 11-43
 Practice Session Overview: Lesson 11 11-45
 Practice Session: Lesson 11 11-46

Lesson 12: Creating and Using Report Parameters

Overview 12-3
 Creating User Parameters 12-5
 Referencing Parameters in a Report Query 12-7
 Hints and Tips for Referencing Parameters 12-13
 Creating a List of Values 12-15
 Referencing System Parameters 12-19
 Building a Parameter Form 12-23
 Summary 12-27
 Practice Session Overview: Lesson 12 12-29
 Practice Session: Lesson 12 12-30

Lesson 13: Embedding a Chart in a Report

Overview 13-3
 Creating and Embedding a Graphics Chart in a Report 13-5
 Embedding an Existing Chart in a Report 13-9
 Summary 13-11
 Practice Session Overview: Lesson 13 13-13
 Practice Session: Lesson 13 13-14

Lesson 14: Enhancing Matrix Reports

Overview 14-3
 The Matrix Data Model 14-5
 Matrix Layout 14-7
 Creating Matrix Summaries 14-9
 Building a Nested Matrix 14-11
 Summary 14-19
 Practice Session Overview: Lesson 14 14-21
 Practice Session: Lesson 14 14-22

Lesson 15: Coding PL/SQL Triggers in Reports

- Overview 15-3
- Types of Triggers in Reports 15-5
- Using Report Triggers 15-7
- Using Data Model Triggers 15-11
- Using Layout Model Triggers 15-15
- Writing Common Code 15-27
- Summary 15-29
- Practice Session Overview: Lesson 15 15-31
- Practice Session: Lesson 15 15-32

Lesson 16: Using the Report Builder Built-In Package

- Overview 16-3
- Contents of the Report Builder Built-in Package 16-5
- Outputting Messages 16-7
- Executing a Nested Report 16-9
- Restricting Data 16-15
- Initializing Fields 16-17
- Performing DDL Statements 16-19
- Setting Format Attributes 16-21
- Summary 16-25
- Practice Session Overview: Lesson 16 16-27
- Practice Session: Lesson 16 16-28

Lesson 17: Maximizing Performance Using the Reports Server

- Overview 17-3
- Why Use a Remote Server? 17-5
- Reports Server Architecture and Configuration 17-9
- Reports Server Clustering 17-11
- Viewing and Scheduling Reports in the Queue Manager 17-25
- Summary 17-33
- Practice Session Overview: Lesson 17 17-35
- Practice Session: Lesson 17 17-36

Lesson 18: Building Reports for Different Environments

Overview	18-3
Different Development and Run-Time Environments	18-5
Developing Reports to Run in Different GUIs	18-7
Building Character Mode Reports	18-9
Building Reports to Run in Different Languages	18-13
Summary	18-17

Appendix A: Solutions

Practice Session Overview: Lesson 1	A-2
Practice Session Solutions: Lesson 1	A-3
Practice Session Overview: Lesson 2	A-8
Practice Session Solutions: Lesson 2	A-9
Practice Session Overview: Lesson 3	A-14
Practice Session Solutions: Lesson 3	A-15
Practice Session Overview: Lesson 4	A-18
Practice Session Solutions: Lesson 4	A-19
Practice Session Overview: Lesson 5	A-24
Practice Session Solutions: Lesson 5	A-25
Practice Session Overview: Lesson 6	A-30
Practice Session Solutions: Lesson 6	A-31
Practice Session Overview: Lesson 7	A-34
Practice Session Solutions: Lesson 7	A-35
Practice Session Overview: Lesson 8	A-37
Practice Session Solutions: Lesson 8	A-38
Practice Session Overview: Lesson 9	A-44
Practice Session Solutions: Lesson 9	A-45
Practice Session Overview: Lesson 10	A-51
Practice Session Solutions: Lesson 10	A-52
Practice Session Overview: Lesson 11	A-57
Practice Session Solutions: Lesson 11	A-58
Practice Session Overview: Lesson 12	A-61
Practice Session Solutions: Lesson 12	A-62

Practice Session Overview: Lesson 13	A-65
Practice Session Solutions: Lesson 13	A-66
Practice Session Overview: Lesson 14	A-69
Practice Session Solutions: Lesson 14	A-70
Practice Session Overview: Lesson 15	A-75
Practice Session Solutions: Lesson 15	A-76
Practice Session Overview: Lesson 16	A-81
Practice Session Solutions: Lesson 16	A-82
Practice Session Overview: Lesson 17	A-90
Practice Session Solutions: Lesson 17	A-91

Appendix B: Table Descriptions and Data

Summit Sporting Goods Database Diagram	B-2
S_CUSTOMER Description	B-3
S_CUSTOMER Data	B-4
S_DEPT Description and Data	B-8
S_EMP Description	B-9
S_EMP Data	B-10
S_ITEM Description	B-13
S_ITEM Data	B-14
S_ORD Description and Data	B-16
S_PRODUCT Description	B-17
S_PRODUCT Data	B-18
S_REGION Description and Data	B-22
S_TITLE Description and Data	B-23
Oracle8 Objects: Types, Tables	B-24

Appendix C: Frequently Asked Questions

Frequently Asked Questions	C-2
Frequently Asked Questions and Answers	C-3

Appendix D: Oracle Rdb Overview

What Is Oracle Rdb?	D-2
---------------------	-----

Appendix E: Introduction to Query Builder

- What Is Query Builder? E-3
- The Query Builder Window E-7
- Building a New Query E-9
- Datasource Components E-11
- Refining a Query E-13
- Viewing and Saving Queries E-17
- Including Additional Tables E-19
- Relationships E-23
- Selecting Rows with Conditions E-29
- Operators E-31
- Entering Multiple Conditions E-33
- Changing Conditions E-35
- Defining Columns Using an Expression E-37
- Defining Columns Using a Function E-39

Appendix F: Object Features in Oracle Developer

- Overview F-3
- New Oracle8 Datatypes F-5
- Creating Oracle8 Objects F-11
- Referencing Objects F-19
- Displaying Oracle8 Objects in the Object Navigator F-21
- Summary F-29

Appendix G: Using the Layout Editor in Oracle Developer

- Overview G-3
- Why Use the Layout Editor? G-5
- How to Access the Layout Editor G-7
- The Layout Editor Components G-9
- Creating and Modifying Objects in the Layout G-11
- Formatting Objects in the Layout G-19
- Coloring Objects and Text G-21
- Importing Images and Drawings G-25
- Summary G-27

Contents

Preface

Profile

Before You Begin This Course

Before you begin this course, you should have:

- Thorough knowledge of creating SQL query statements
- Working experience of:
 - Creating PL/SQL constructs, including conditional statements, procedures and functions.
 - Creating PL/SQL stored (server) procedures and functions
 - Using a graphical user interface (GUI)

Prerequisites

- The following instructor-led training (ILT) courses:
 - *Introduction to Oracle: SQL & PL/SQL* or *Introduction to Oracle for Experienced SQL Users*
 - *Develop PL/SQL Program Units*
- Or the following computer-based training (CBT) courses from the SQL * PL/SQL CBT Library:
 - *SQL and SQL*Plus: Basic SELECT Statements*
 - *SQL and SQL*Plus: DDL and DML*
 - *PL/SQL: Basics*
 - *PL/SQL: Procedures, Functions sand Packages*
 - *PL/SQL: Database Programming*

Suggested

- *Advanced SQL and SQL*Plus* (ILT)
- Or *SQL and SQL*Plus: Advanced SELECT Statements* (CBT)

Suggested Follow-up Courses

- *Oracle Developer: Build Forms I*
- *Oracle Developer: Deploy Web-Based Applications*
- *Oracle Designer: Design and Generate Oracle Developer Applications*

How This Course Is Organized

- *Oracle Developer: Build Reports* is an instructor-led course featuring lecture and hands-on exercises. Online demonstrations and written practice sessions reinforce the concepts and skills introduced.

Related Publications

Oracle Publications

Title	Part Number
<i>Oracle Developer: Guidelines for Building Applications Release 6.0</i>	A58766-02
<i>Oracle Developer Getting Started Release 6.0</i>	A58767-02

Additional Publications

- Release notes PDF format: <oracle_home>\tools\doc60\relnotes.pdf
- Release notes text format: <oracle_home>\tools\doc60\relnotes.txt
- System release bulletins
- Installation and user's guides
- *read.me* files
- International Oracle User's Group (IOUG) articles
- *Oracle Magazine*

Typographic Conventions

Typographic Conventions in Text

Convention	Element	Example
Bold italic	Glossary term (if there is a glossary)	The <i>algorithm</i> inserts the new key.
Caps and lowercase	Buttons, check boxes, triggers, windows	Click the Executable button. Select the Can't Delete Card check box. Assign a When-Validate-Item trigger . . . Open the Master Schedule window.
Courier new, case sensitive (default is lowercase)	Code output, directory names, filenames, passwords, pathnames, URLs, user input, usernames	Code output: <code>debug.seti('I',300);</code> Directory: bin (DOS), \$FMHOME (UNIX) Filename: Locate the <code>init.ora</code> file. Password: Use <code>tiger</code> as your password. Pathname: Open <code>c:\my_docs\projects</code> URL: Go to <code>http://www.oracle.com</code> User input: Enter <code>300</code> Username: Log on as <code>scott</code>
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address (<i>but</i> Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	<i>Do not</i> save changes to the database. For further information, see <i>Oracle7 Server SQL Language Reference Manual</i> . Enter <i>user_id</i> @us.oracle.com, where <i>user_id</i> is the name of the user.
Quotation marks	Interface elements with long names that have only initial caps; lesson and chapter titles in cross-references	Select "Include a reusable module component" and click Finish. This subject is covered in Unit II, Lesson 3, "Working with Objects."
Uppercase	SQL column names, commands, functions, schemas, table names	Use the SELECT command to view information stored in the LAST_NAME column of the EMP table.

Convention	Element	Example
Arrow	Menu paths	Select File—>Save.
Brackets	Key names	Press [Enter].
Commas	Key sequences	Press and release these keys one at a time: [Alt], [F], [D]
Plus signs	Key combinations	Press and hold these keys simultaneously: [Ctrl]+[Alt]+[Del]

Typographic Conventions in Code

Convention	Element	Example
Caps and lowercase	Oracle Forms triggers	When-Validate-Item
Lowercase	Column names, table names	SELECT last_name FROM s_emp;
	Passwords	DROP USER scott IDENTIFIED BY tiger;
	PL/SQL objects	OG_ACTIVATE_LAYER (OG_GET_LAYER ('prod_pie_layer'))
Lowercase italic	Syntax variables	CREATE ROLE <i>role</i>
Uppercase	SQL commands and functions	SELECT userid FROM emp;

Typographic Conventions in Navigation Paths

This course uses simplified navigation paths, such as the following example, to direct you through Oracle Applications.

(N) Invoice—>Entry—>Invoice Batches Summary (M) Query—>Find
(B) Approve

This simplified path translates to the following:

- 1 (N) From the Navigator window, select Invoice—>Entry—>Invoice Batches Summary.
- 2 (M) From the menu bar, select Query—>Find.
- 3 (B) Click the Approve button.

N = Navigator, **M** = Menu, **B** = Button

I

Introduction

Objectives

At the end of this section, you should be able to do the following:

- **Identify the course objectives**
- **Identify the course content and structure**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

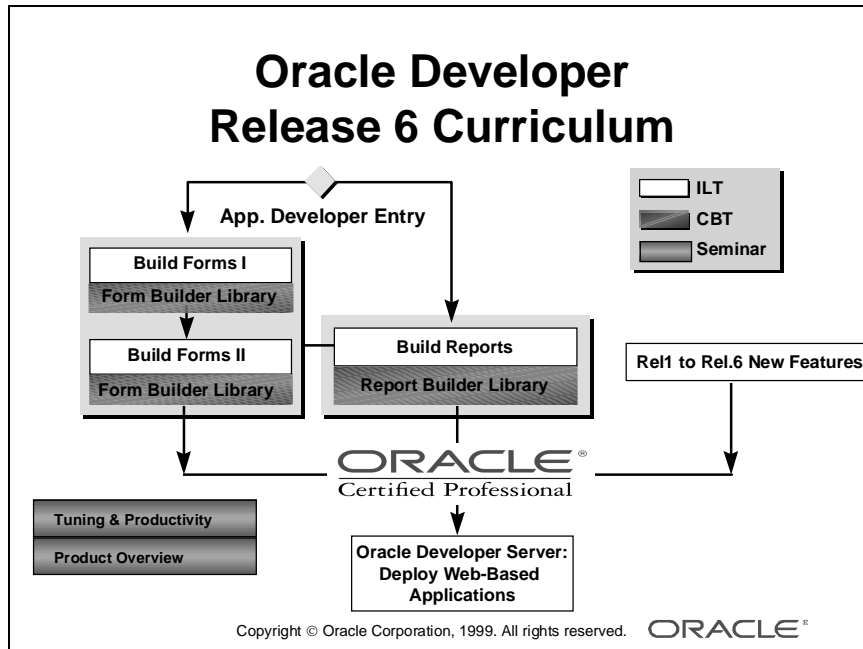
Introduction

This lesson introduces you to the course: the objectives that the course intends to meet, the topics that it covers, and how the topics are structured over the duration of the course.

Objectives

At the end of this lesson, you should be able to do the following:

- Identify the course objectives
- Describe the course content and structure



Course Objectives

At the end of this course, you should be able to do the following:

- **Describe the capabilities of Oracle Developer Reports**
- **Develop and maintain different styles of reports using Oracle Developer Report Builder**
- **Lay out and format data to suit user reporting requirements**
- **Incorporate text and graphics in reports**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Course Objectives

Course Description

This course enables you to create reports in a variety of styles, customizing them to meet specific business requirements by combining text and graphics with quantitative data.

During the course, you develop reports in a client-server environment using the power and flexibility of the Oracle Developer graphical user interface (GUI).

Course Objectives

By the end of this course you should be able to:

- Describe the capabilities of Oracle Developer Reports
- Develop and maintain different styles of reports using Oracle Developer Report Builder
- Lay out and format data to suit user reporting requirements
- Incorporate text and graphics in reports

Course Content

Day 1—Basic concepts

- Introducing Oracle Developer
- Designing and running reports
- Creating and modifying reports using the wizard

Day 2—Enhancements

- Managing templates
- Enhancing report structure and layout

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Course Content

Day 3—More enhancements

- Customizing reports with parameters
- Embedding a chart
- Adding PL/SQL code

Day 4—Additional features

- Using built-in packages
- Scheduling remote reports
- Building reports for different environments

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Course Content

The lesson titles show the topics covered in this course and the usual sequence of lessons. However, the daily schedule is an estimate, and may vary for each individual class.

Day 1

Course Introduction

Introduction to Oracle Developer

Designing and Running Reports

Report Builder Concepts

Creating Reports Using the Report Wizard

Enhancing Reports in the Live Previewer

Day 2

Managing Report Templates

Report Storage Methods

Enhancing Reports Using the Data Model: Creating Queries and Groups

Enhancing Reports Using the Data Model: Creating Columns

Enhancing Reports Using the Layout Model

Day 3

Modifying Properties

Creating and Using Report Parameters

Embedding a Chart in a Report

Enhancing Matrix Reports

Coding PL/SQL Triggers in Reports

Day 4

Using the Report Builder Built-In Package

Maximizing Performance Using the Reports Server

Building Reports for Different Environments

1

Introduction to Oracle Reports

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe common features of Oracle Developer**
- **Describe Oracle Developer components**
- **Describe common builder components**
- **Navigate around the Report builder interface**
- **Customize the Oracle Reports session**
- **Describe the course application**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

This course teaches you to build effective and professional reports using Oracle Reports. This is only one component of Oracle Developer, and it is useful for you to understand the whole suite of components and the capabilities of Oracle Developer in building consistent, integrated applications.

This lesson identifies the key and common features of Oracle Developer, the Oracle Developer components, the Oracle Reports interface, and the course application model and contents.

Objectives

At the end of this lesson, you should be able to do the following:

- Describe the common features of Oracle Developer
- Describe the Oracle Developer components
- Describe the common builder components
- Navigate around the Oracle Reports interface
- Customize the Oracle Reports session by modifying builder preferences
- Describe the course application

What Is Oracle Developer?

- A suite of Oracle components with common interfaces
- One of a family of products

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

What Is Oracle Developer?

Family of products



Oracle Designer



Oracle Developer

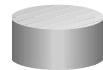


Oracle Discoverer

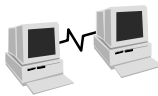
Supported technologies



GUI



Database



Client-Server



World Wide Web

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

What Is Oracle Developer?

Introduction

Oracle Developer is a suite of Oracle components that you can integrate to build comprehensive applications.

Family of Products

Oracle Developer is one group within a family of products that support the design, creation, and running of applications across a variety of platforms.

Oracle's design, development, and data access tools are:

- Oracle Designer
- Oracle Developer
- Oracle Discoverer

Development Features

Oracle Developer lets you build high-performance systems that take advantage of GUI, database, client-server, and Web technologies.

The intuitive tools in Oracle Developer have common interfaces and features that make it easy to move between tools and to design elements that are both modular and reusable.

Key Features

- **Comprehensive GUI support**
- **Distributed applications**
- **Productive and comprehensive tools**
- **Application partitioning**
- **Flexible source control**
- **Extended scalability**
- **Object orientation**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Oracle Developer Key Features

Oracle Developer Release 6.0 provides a number of features that contribute to its strength and flexibility.

Comprehensive GUI Support Oracle Developer supports the native features of Microsoft Windows95 and NT 4.0, and provides portability to UNIX, Macintosh, and character-mode production environments.

Distributed Applications The tools provide local, client-server, and Web support with multiple database connections per application. As well as Oracle7 and Oracle8, your applications can access SQL databases through ODBC.

Productive and Comprehensive Tools Oracle Developer provides the same design facilities across tools, for common functions such as the Layout editor and the Object Navigator.

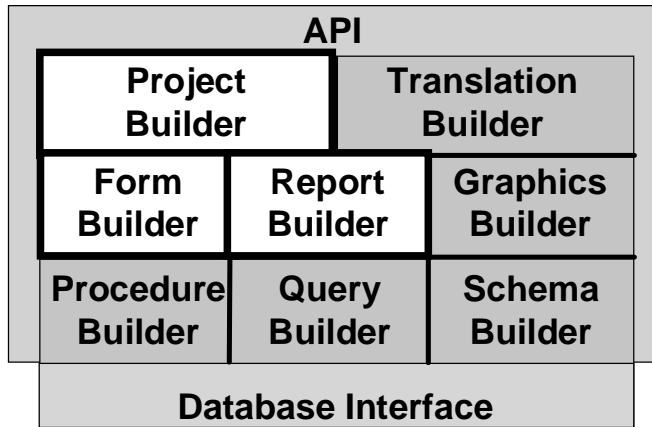
Application Partitioning You can place individual PL/SQL program units on the database server, on the application server, or in the client-side application, whichever is most suitable in each case. You can copy and move objects between modules and database server, using convenient drag-and-drop techniques.

Flexible Source Control You can store the definitions of your application modules in flat files or in the Oracle database. You can perform version control on these modules and produce documentation through Oracle Developer facilities.

Extended Scalability You can scale applications from single users to tens of thousands, with no changes to the application. Scalability is inherent in the multitiered architecture of the product. There is support for server functionality, such as array DML, database cursors, bind variables, and result sets.

Object Orientation Oracle Developer offers an inheritance model that facilitates the inheritance of attributes and code from one object to another and from one application to another, through subclassing and object libraries.

Oracle Developer Components



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Introducing the Oracle Developer Release 6.0 Components

Project Builder

This component helps in the entire project life cycle. A navigator-style interface provides easy access to all your project files and the actions associated with them. The customizable launcher enables you to launch Oracle and non-Oracle applications, such as Form Builder, Report Builder, and MS-Word. You can package your application and then deploy it on other machines using utilities of your own choosing.

Form Builder

This development interface helps you build sophisticated, interactive applications—screens and pull-down menus. Form Builder can present information through textual items, GUI objects, and bit-mapped images. Users can perform database transactions by interacting with these objects.

Wizards help you build simple, standard modules very quickly. You can then use the powerful features of Form Builder to enhance the appearance and functionality of your application.

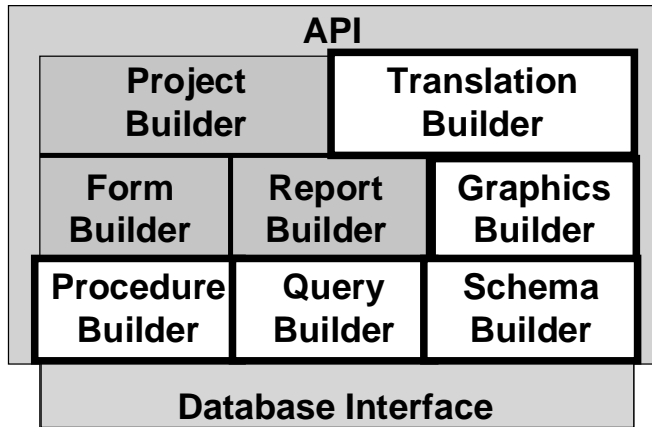
Report Builder

This development interface helps you build both simple and extremely complex production-quality reports. The wizard offers a variety of styles. Templates help you develop professional, standard reports quickly and easily, and provide reusability and a standard look and feel. Report Builder also supports many more complex and individual enhancements.

You can run a report as a stand-alone application, or invoke it from an Oracle Developer form, passing data to the report as required. You can deploy reports on the Web using the Reports Server and Reports Web CGI or Reports Web cartridge for Oracle Application Server. Oracle Reports Application Server based architecture means you only need a Web browser to view reports in industry standard formats such as HTML and PDF. Scalability is guaranteed through the fully distributed, clustered Reports Server.

Reports is integrated with Oracle Discoverer. Oracle Discoverer end users can slice and dice looking for answers to add-hoc questions. These answers can be instantly transformed from static worksheets to dynamic Web reports. The integration with Oracle Express Server allows users to build reports against data held within a multidimensional database. These reports can then seamlessly be published over the Web.

Oracle Developer Components



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Introducing the Oracle Developer Release 6.0 Components (continued)

Graphics Builder

This development interface helps you build a variety of charts, including drill-down graphics applications. Use the Graphics Builder to enhance and modify charts created by the Chart wizard in the Form and Report Builders.

Note: You can automatically generate Oracle Developer applications from design specifications by using Oracle Designer.

Query Builder

With this user-friendly graphical interface you can build queries, either in a stand-alone executable or during development of a report using Report Builder.

Schema Builder

With this user-friendly graphical tool you can define database objects and their relationships in a stand-alone executable.

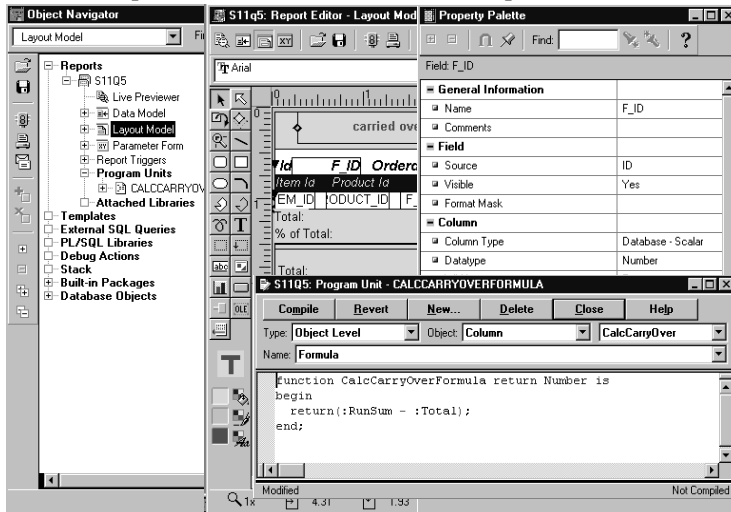
Procedure Builder

Procedure Builder provides the common PL/SQL and program unit management facilities contained in the Oracle Developer application builders (Form, Report, and Graphics). You can also invoke the stand-alone builder, which allows you to create and manage program units in a separate session, without invoking the other application builders.

Translation Builder

This user-friendly and powerful tool supports and manages translations of text extracted from Oracle resources. It is a cost-effective solution to translate a given application, and it helps Oracle developers to make a simultaneous release of native and multilingual applications.

Report Builder Components



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report Builder Components

Object Navigator

The Object Navigator is a hierarchical browsing and editing interface that enables you to locate and manipulate application objects easily and quickly. Features include:

- A hierarchy represented by indentation and expandable nodes. (Top-level nodes show module types, database objects, and built-in packages.)
- Find field and icons, enabling forward and backward searches for any level of node or for an individual item in a node
- Icons in the vertical toolbar replicating common File menu functions

Layout Model

The Layout Model editor lets you design the style, color, size, and arrangement of visual objects in the application. These layouts can include graphical objects and bitmapped images when running in a GUI environment.

Property Palette

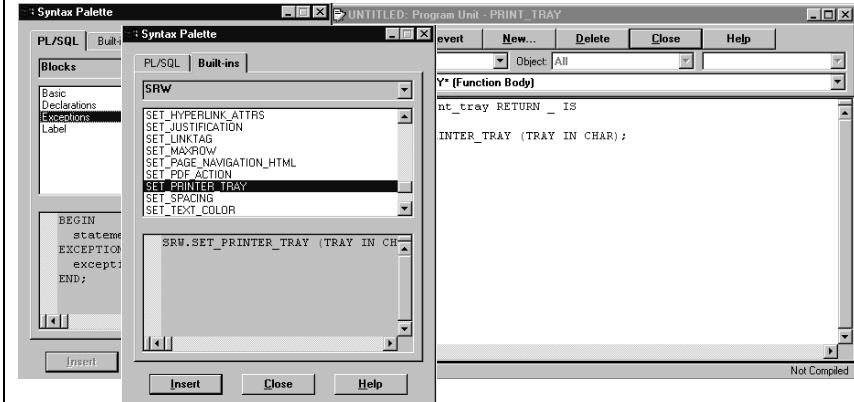
All objects in a module, including the module itself, have properties that you can see and modify in the property palette. Features include:

- Copy and reuse properties from another object
- Find field and icons, similar to Object Navigator

PL/SQL Editor

The PL/SQL Editor lets you create and compile program units such as functions, procedures and packages within the current report.

PL/SQL Development Environment: Syntax Palette



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

PL/SQL Development Environment

The Development Environment is the integrated functionality of Procedure Builder that exists within the application builders. It provides:

- Development of server-side database triggers and stored procedures, functions, and packages
- Development of libraries to hold PL/SQL program units
- Statement-level debugging of PL/SQL at run time

Syntax Palette

The Syntax Palette is a programming tool that enables you to display and copy the constructs of PL/SQL language elements and built-in packages into the PL/SQL editor.

Getting Started

- **Starting the builders:**
 - Invoke individual builder icons
 - Invoke from Oracle 6.0 group
- **What you see**
 - Builder Wizard
- **Connecting to the database**
 - File→Connect

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Getting Started in the Oracle Reports Interface

Starting the Builders

You invoke an individual builder from the Oracle Developer 6.0 group.

What You See in the Report Builder

When you invoke the Report Builder, you first see the wizard. The wizard gives you several options, including:

- Create a new module using the wizard
- Create a new module manually
- Open an existing module

Database Connection

If you build applications that access database objects, you need to connect to a database account from the builder.

Connect to a database if you need:

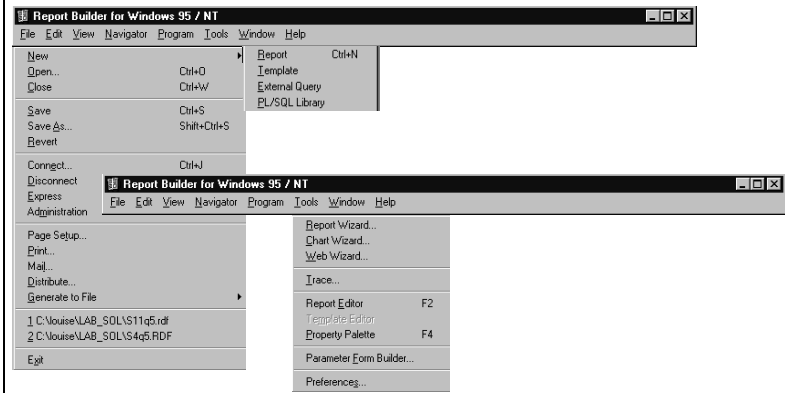
- To compile code that contains SQL
- To access database objects in the Object Navigator
- To create Oracle Developer objects that are based on database objects

How to Connect to Oracle

- 1 Select File—>Connect from the menu.
- 2 Enter the database user and password in the Connect dialog box. If you are not connecting to the default database, also provide the necessary connect string or database alias.

Note: Oracle Reports automatically displays the Connect dialog box if you try to perform a task that requires connection.

Main Menu Structure



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Navigating Around the Oracle Reports Main Menu

Introduction

The main menus in the three main application builders (Form, Report, Graphics) are very similar. The main menu contains options in one builder that do not always appear in another, but their basic functionality is the same throughout. The main menu also varies depending on your current context in the builder.

Common Menu Features

The following table describes some common features in GUI menus.

Feature	Description
Underline	Shortcut key: [Alt]+letter
Ellipsis (...)	Additional input, usually by dialog box
>	Menu option has a submenu
Windows menu	List of open windows; choose any window to make it active
Help	List of help facilities, such as online help text, Cue Cards, Quick Tour

Native GUI Interface

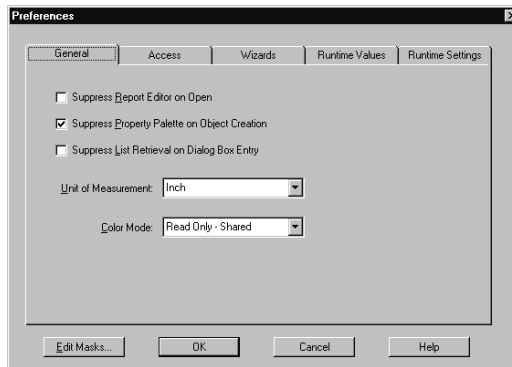
The menu example here is shown in Windows NT. However, menus appear with the look and feel of your native GUI interface.

Example In Motif, the Windows Print dialog box options appear as submenus of the Font menu.

The Builder Main Menu

Menu Item	Description
File	Common file utilities, such as open, save, connect, administration
Edit	Cut, copy, paste, and so on
View	Switch view in current window; options vary greatly depending on context
Navigator	Appears only in Object Navigator; includes expand and collapse, bookmark
Program	Includes compilation and access to PL/SQL editors
Tools	Includes wizards and preferences

Customizing your Oracle Reports Session



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Customizing Your Oracle Reports Session

What Are Report Builder Tools Preferences?

Tools preferences allow you to customize some aspects of your Oracle Report Builder session.

Report Preferences

There are five tab pages in the Report Preferences dialog box. Press the Help key ([F1] for Windows NT/95) in the Preferences dialog box to see a description of each preference.

As well as session preferences, Report Builder preferences enable you to set run-time settings when running your report within the builder.

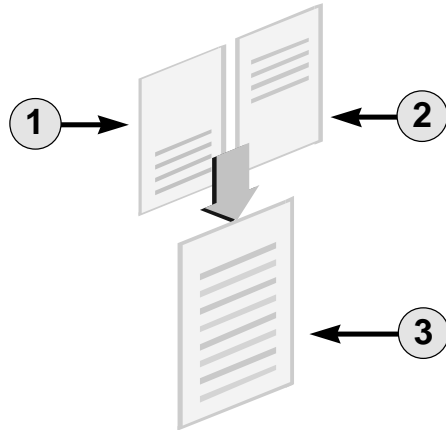
The following table describes a few example preferences. Others are introduced when applicable throughout the course.

Tab	Preference Name	Description
General	Suppress Report Editor on Open	Do not display the Report editor when opening a report. This saves you time when opening several reports to make changes in the Object Navigator.
	Unit of Measurement	Set the unit of measurement that you want to use for new reports that you create. Altering this setting does not affect existing report definitions.
Access	Access	Determines if report definitions are saved to (or opened from) the database or from files, or both. If access is set to both, Report Builder prompts you when you save or open each report.
Wizards	Welcome Dialog	Check box to suppress or display the first Welcome dialog box. There are several similar check boxes.

How to Modify Tools Preferences

- 1 Select Tools—>Preferences.
- 2 Select the option you require.
- 3 Choose OK to save changes or Cancel to cancel changes.

Saving Tools Preferences



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Existing preferences file
2	Modified preferences
3	Updated, merged preferences file

Saving Tools Preferences

When you choose OK in the Preferences dialog box, Oracle Reports updates your current session with the changes. When you exit the builder, Oracle Reports writes the changes to a preference file for future sessions.

If the preference file already exists, changes are merged with the existing file. This means that preferences for other tools are not affected.

Each option in the preference file is prefixed by the Oracle Developer tool name to which it belongs.

Example

```
OracleGraphics.Rulers = Yes
Reports.Object_Access = File
Reports.Show_All = Yes
Forms.welcome_dialog = on
```

Oracle Reports reads the preference file whenever you invoke the Report Builder.

The name of the preference file varies on different platforms:

Window Manager	Preference Filename
Microsoft Windows	cauprefs.ora
Motif	prefs.ora

Note: The preferences file is an editable text file. However, where possible, you should alter the options in the Preferences dialog box.

Oracle Reports Environment Variables

- **REPORTS60_PAT**
- **UI_ICON**
- **ORACLE_PATH**

Windows NT/95: modify in Registry

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**®

Oracle Reports Environment Variables

Introduction

Oracle Reports uses many environment variables. These have default values, all of which you can modify in your own environment and for different applications. This section discusses only those variables that relate to file searching at run time.

Setting Pathnames

Report Builder uses some environment variables to search for files at run time. This enables you to build applications that are portable across platforms and directory structures by avoiding hardcoded paths in file references in a form, report, or chart.

Variable	Description
REPORTS60_PATH	A path that Report Builder searches for files at run time
UI _ICON	A path that all builders search for icon files at run time

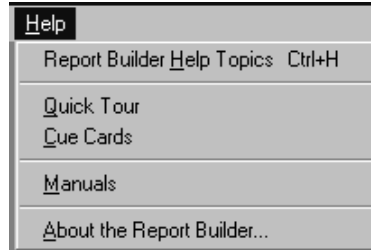
Generic Oracle Path

ORACLE_PATH is an additional path that all components search if they cannot find a file in their own specific path.

Modifying Environment Variables

In a Windows 32-bit environment, use the Windows Registry to modify these paths. Registry path: HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE.

Using the Online Help System



- **Quick Tour: built-in training package of technical information (HTML files)**
- **Cue Cards: simple step-by-step instructions**

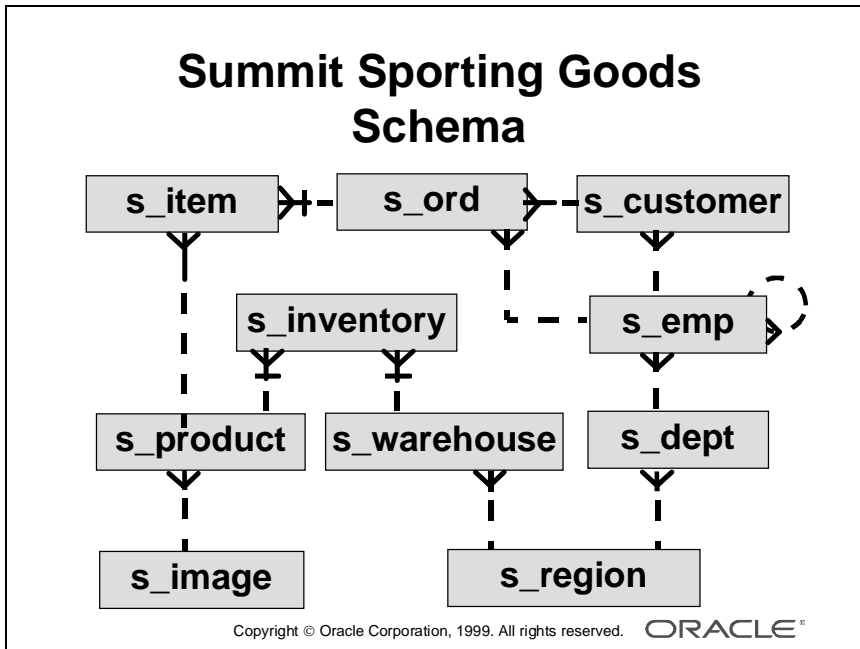
Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Oracle Reports Help Options

The table describes the Help menu options in Report Builder.

Help Menu Option	Description
Help Topics	<p>This is the Contents Page for comprehensive online help. Includes Index and Find tabs.</p> <p>The Help key ([F1] for Windows NT/95) displays context-sensitive online help at any place in the builder.</p>
Quick Tour	<p>This built-in training package of HTML files is more detailed than the Cue Cards, including technical explanations of the Reports components.</p>
Cue Cards	<p>These appear as separate windows that give simple step-by-step instructions and examples to help novice users learn basic functionality.</p>
Manuals	<p>This provides an index to installed online manuals, which are HTML files that you can view with any browser.</p>
About the Report Builder	<p>Appears as two panels:</p> <p>The upper panel shows a server-side connection when you are connected to a database server. If there is no connection, this panel is blank.</p> <p>The lower panel shows client-side components and their version numbers.</p>

Most of the Help features are optional extras during product installation, and they require additional resources.



Introducing the Course Application

The Summit Sporting Goods Schema

The simplified table diagram shows the tables used throughout this course to build reports. You may be familiar with these tables from other Oracle courses.

Summit Sporting Goods is a company that supplies sports goods to customers. Summit has a number of employees in several departments. Some employees are sales representatives who have a relationship to specific customers.

Customers place orders. Each order consists of one or more line items. Each line item represents a product.

Each product has an image associated with it, in the form of an image file.

The company products are stored in a number of warehouses. The contents of the warehouses are managed in the inventory.

Summary

- **Oracle Developer common features**
- **Report Builder interface**
- **Report Builder preferences**
- **The SUMMIT course application**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

- Oracle Developer provides a suite of components with common features and a common builder interface for forms, reports, and graphic charts.
- The product provides a common builder interface, including Object Navigator, Layout editor, PL/SQL editor, and property palette components, and offers a comprehensive online Help system.
- Oracle Reports has a set of preferences that you can alter for the current and subsequent builder sessions.
- Finally, this lesson described the SUMMIT application tables that are used throughout the course to build reports.

Practice 1 Overview

- **Navigating through the Oracle Reports interface**
- **Investigating Help topics**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 1

Introduction

This practice familiarizes you with Oracle Reports by asking a number of questions that you answer by navigating through the interface and by investigating Help Topics. After each question, space is provided for you to write the answer. Or, if you prefer, use a separate sheet of paper.

Depending on the size of your group, the instructor might ask you to answer some of the questions and share your answers with the rest of the class.

Depending on the environment you are using in this course, the instructor might ask you to compile some reports from the Project Builder.

Practice Session: Lesson 1

- 1 Launch Report Builder and choose Cancel in the wizard.
- 2 Change your preferences so that a newly opened report is not immediately executed. To do this you must suppress the Report editor on Open.
- 3 Open the report named `p2q2.rdf`.
- 4 Change your preferences so that you can work with a report definition either in a file or in the database. Ensure that the default list shows only reports.
- 5 Open another report: `p2q3.rdf`. Notice the additional dialog box.
- 6 Reset your preferences to work with files only. Enable a report to be run automatically on open.
- 7 Use the context-sensitive Help to discover more about preferences.

General:

- a What setting must you choose for color mode if you want to alter the color palette for a specific report?
- b What menu option do you use to alter the color palette?

Run-time values:

- c What is the purpose of the run-time values?
- d When do they take effect?
- e What must you do to change the default setting in your report?
- f What are the names of the three run-time values that affect the report destination, and what are the three related report parameters?

8 Use Help Topics to discover more about the functionality of Oracle Developer Report Builder:

- a** What is MAPI?
- b** What are the prerequisites for sending a report as an e-mail attachment?
- c** The report is sent as what type of attachment format?
- d** What are the three steps to install Oracle Developer demonstrations?

9 Use Help on Manuals to answer the following questions in the Oracle Information Navigator:

Note: In the Information Navigator, choose Tools—>Preferences to initialize your Web browser. Your instructor will tell you which browser to use.

- a** What are the three top-level folder names for the online manuals?
- b** In the *Getting Started* manual, what are the five topics included in Administrative Procedures?
- c** In *Designing Portable Applications*, what is the purpose of section 6.3?
- d** Name the three language environment variables available when developing multilingual applications.
- e** What is the Web wizard?
- f** How can you dynamically deploy a report on the Web?

- 10** Use Quick Tour to find out how to prepare a report for the World Wide Web.
- a** What changes do you have to make to your report to deploy it on the Web?

 - b** What output formats can you choose from?

Designing and Running Reports

Objectives

At the end of this lesson, you should be able to do the following:

- **Name the common report styles**
- **Describe the structure of each style**
- **Run prebuilt reports using the Run-time executable**
- **Distinguish between the various report output destinations**
- **View report output in the Previewer**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

Your reports need to satisfy the requirements of your business. To help you begin the process of translating report requirements into Reports solutions, you need to understand the potential range of report styles and gain an appreciation of the underlying structure. This helps you to make the right decision about which style to use for the report.

This lesson discusses examples of standard report design styles and describes how to run existing reports to various output destinations.

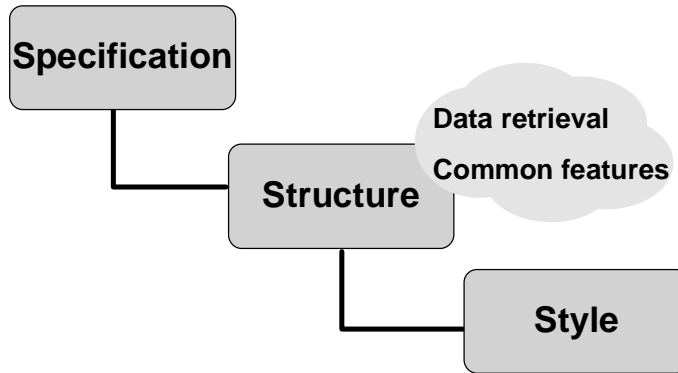
Objectives

At the end of this lesson, you should be able to do the following:

- Name the common report styles required in a business report
- Describe the structure of common report styles
- Run prebuilt reports using the Reports Runtime executable
- Distinguish between the various report destinations
- View report output in the Previewer

Designing Reports

Before you start development, consider:



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Designing Reports

Introduction

Before you create a report using any report-writing software, you must first consider the type of reports that you are being asked to produce. You will have a specification of the required output, but you also need to know the structure that is required and the most efficient way to retrieve data.

Also, consider whether this is a one-time requirement, or whether this specification shares common features with other reports, especially where multiple reports are required in the same application.

There are a few standard styles of reports that form the majority of all reporting requirements. This section teaches you to recognize the common styles in order to enable you to interpret your report requirements and choose the correct style when developing report definitions.

The Common Report Styles

The majority of report requirements fall into the following categories:

- Tabular
- Master-detail
- Master and multiple details
- Matrix

Tabular

List of Products

Product Number	Description	Price



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Design Considerations

Introduction

Define the report structure by identifying the number of times each piece of data is printed. Separate the data into groups based on this frequency. You may also need to identify any relationships that exist between groups, so that you can create a *hierarchy* in the internal report structure.

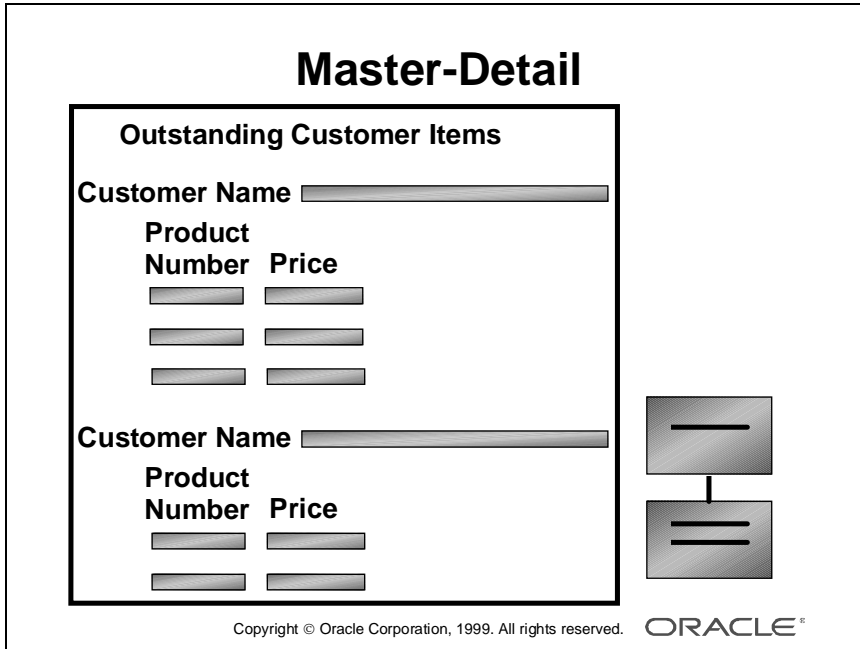
This section describes the four main styles of report and the group structure for each.

Tabular Report Style

In the List of Products report displayed opposite, all fields for each product number, product description, and price are printed with the same frequency. They each repeat a value for every product record.

This report structure contains a single group.

In Tabular reports, the headings or labels appear once above each field.



Master-Detail Report Style

The Outstanding Customer Items report displays one customer name, followed by several records containing the products and prices of items ordered but not yet received. This structure is repeated for every customer who has unfulfilled orders. The customer name does not repeat for every outstanding product; the frequency is different.

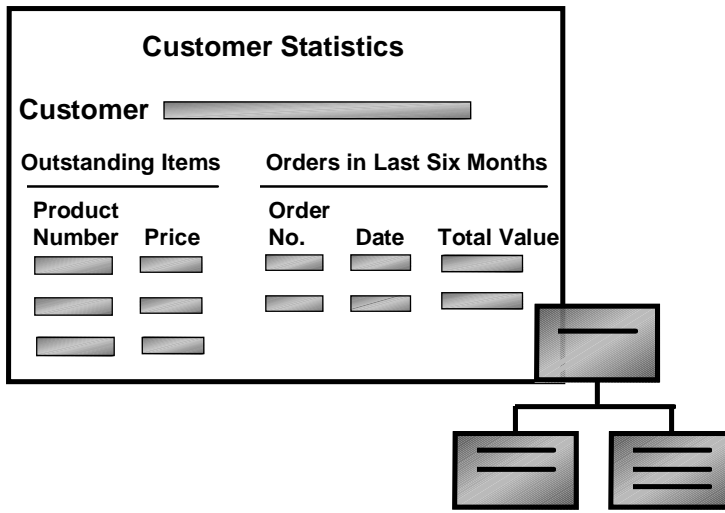
This report structure contains two groups.

Each list of products is directly related to the preceding customer name, which means that there is a master-detail hierarchy. Customer information is in one group, called the *master*, and the product information is in a second group, called the *detail*.

Note: In this Master-Detail report, the headings or labels appear as follows:

Group	Label Placement
Master	Labels appear to the left of fields.
Detail	Labels appear above fields (as for a tabular report).

Master with Two Details



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

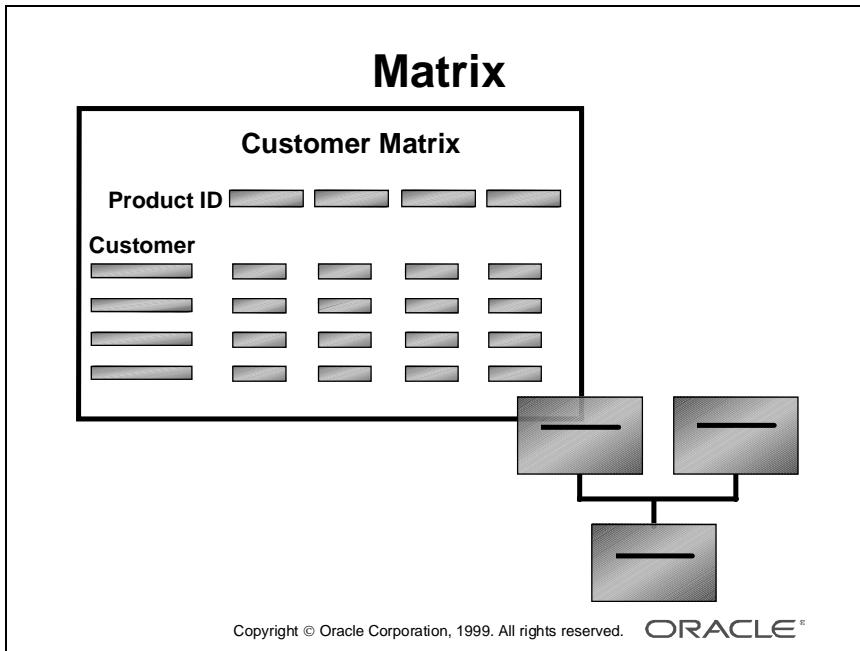
Master with Two Details Style

More than one group may appear at the same level in a hierarchy.

Consider the Customer Statistics report. This report has the same information as the Outstanding Customer Items report, with the addition of another group displaying orders placed in the last six months.

Both groups, Outstanding Items and Orders in Last Six Months, relate to each customer, but not directly to each other.

This report has three groups, one master and two detail groups. The two detail groups are related to the master at the same lower level.



Matrix Report Style

The Customer Matrix report displays a group of master customer records down the page, a group of master product records across the page, and a group of detail summary calculations.

The detail group is related to both of the master groups.

In this report structure, the master groups are on the same level, with the detail group below.

This is an example of a simple matrix. Later you see that matrix reports can have multiple levels of nesting.

Two Masters

The matrix style is the only style in which a detail group is related to two different master groups at the same level.

Retrieving and Sharing Data

- **Keep database access to a minimum**
- **Consider report structure and number of queries**
- **Make effective use of common code and objects**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Retrieving and Sharing Data

Introduction

One of the primary considerations of any report is to ensure that it runs efficiently. The following points provide guidance for future reference.

SQL and Database Access

Data for the report is retrieved from the database using SQL SELECT statements. Aim to keep database access to a minimum. The fewer queries your report contains, the faster it retrieves data.

With hierarchical reports that contain more than one group, you can either use one query and create additional groups, or use many queries and manually link the groups. The one-query approach is usually preferable.

Report Structures

The report style also affects the number of queries required. A master with multiple detail groups requires an additional query for each extra “sibling” group. Matrix reports can contain one or several queries.

The following table shows the number of queries you can use to create the reports in this lesson.

Report Name	Number of Queries
List of Products	1
Outstanding Customer Items	1 or 2
Customer Statistics	2 or 3
Customer Matrix	1, 2 , or 3

Using Common Code

In Oracle Reports, you can create queries, PL/SQL libraries, and stored program units that are accessible to more than one report. Consider which code is common and ensure that all developers have access to it.

Using Common Objects

Many reports have a standard layout for features such as page size, margin widths, company logo image, font size, and style. Consider creating one template containing these common features that can be the basis of all reports. You can also enhance individual reports with specific changes. Good template design standardizes the appearance of your reports and helps to ensure that, as a developer, you maintain documented standards.

Running a Report

Use the RWRUN60 executable from:

- Front end menu
- Menu integrated with a Forms application
- Button in a Forms application
- Command line (Start—>Run)
- Run-time option in your Windows menu
 - Connect to database
 - Enter parameters

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Running a Report

Introduction

To run a report created with Oracle Reports, you use the executable program called RWRUN60.

Invoking a Report

There can be many ways of invoking a report, depending on the application design. You can call a report from:

- A front-end menu
- A customized menu in a Forms application
- A button in a Forms application
- The command line, using the Start—>Run option in Windows NT
- The Reports Runtime item in your Windows menu

Whichever method you use, RWRUN60 is the command that is executed behind the scenes.

How to Run a Report from the Microsoft Windows NT Menu

- 1 Choose the Reports Runtime item from the Oracle Developer R6.0 menu.
- 2 You must connect to the database. In the File—>Connect dialog box enter your username, password, and database alias. Choose Connect.
Note: If you are connecting to your default database, you do not need to specify the alias.
- 3 Choose File—>Run. In the file selection dialog box search through the list of reports, select the report you want to run, and choose OK. You may need to change your drive or directory. If so, amend the Drives list.
The Runtime Parameter Form is displayed for some reports. This allows you to enter different run-time values, such as the report destination.
- 4 Enter the parameter values required and choose Run Report. You can cancel the report by choosing Cancel.

Report Destinations



Screen: format with screen fonts



Preview: format with printer fonts



File: choose file format



Mail: MAPI-compliant



Printer: output direct to printer

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Viewing the Report Output

Introduction

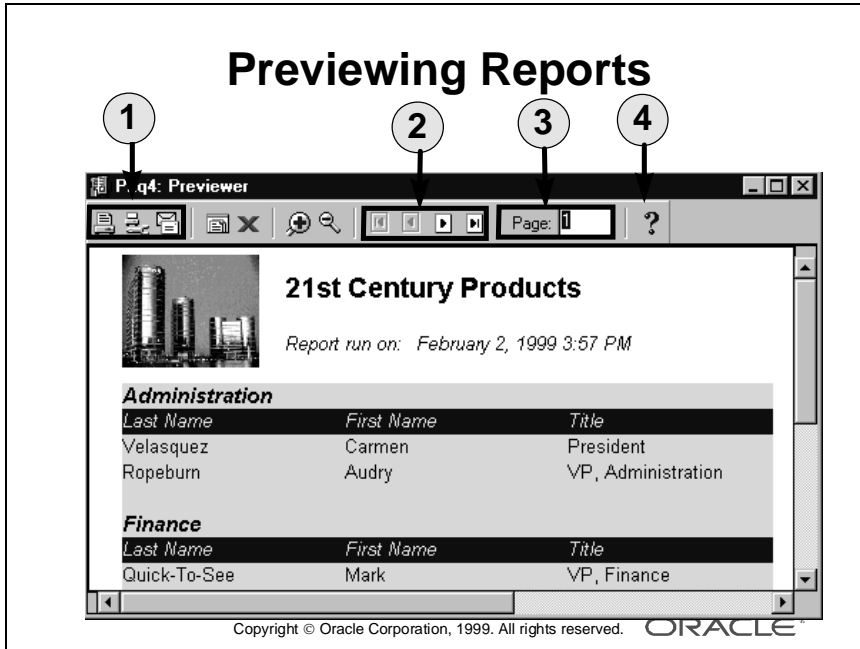
You use the Destination Type parameter to control whether you see the report on the screen, send it to a printer, or output it to a file.

Report Destination Options

There are a number of options available to the user.

Destination Type	Actual Destination
Screen	Routes the output to the Previewer for interactive viewing. Useful when viewing output on the screen only.
Preview	Routes the output to the Previewer for interactive viewing when users require subsequent printing of the output. However, Previewer causes the output to be formatted using a printer font. The Previewer uses Destination Name to determine which printer's fonts to use to display the output.
File	Saves the output to a file named in Destination Name. The format depends on the Destination Format; this can be a printer definition or a browser format, such as PDF and HTML.
Printer	Routes all output directly to the printer named in Destination Name.
Mail	Sends the output to a user of Oracle Mail, Oracle Office, Oracle InterOffice, or any MAPI-compliant mail system. Username is specified in Destination Name.

Note: A report can be output to multiple destinations in one single run of the report.



1	Print, Print Setup, and Mail options	3	Go to specific page number
2	Page options: First, Previous, Next, Last	4	Help: Previewer

Previewing Reports

When you specify a destination of Screen or Preview, your formatted report output appears in the Report Previewer.

You can navigate the screen, view each page, and even split the screen and scroll separately through each portion.

The Previewer Window

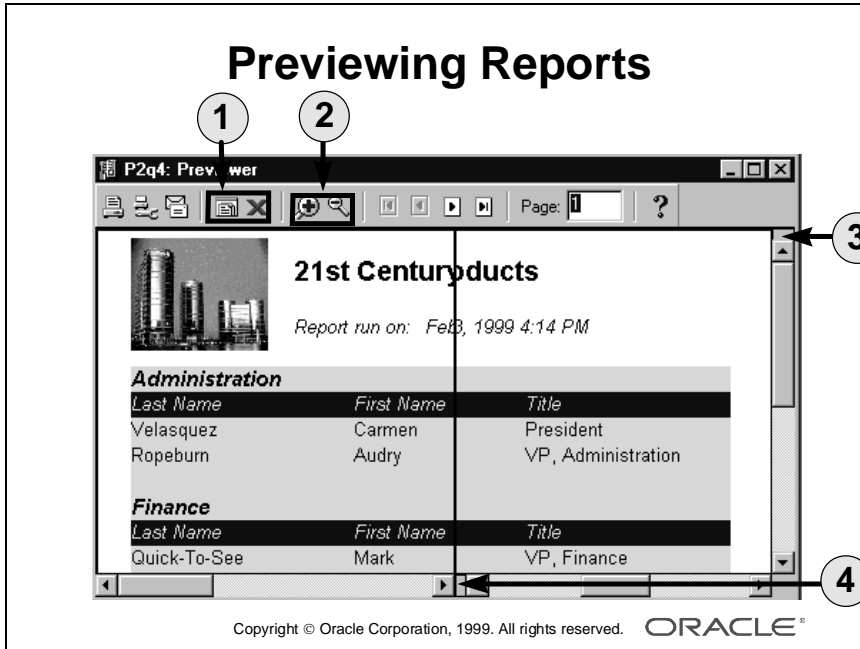
The Previewer has three components:

Component	Description
Main Menu	Contains a subset of menu options, such as Quit and Help
Title	Default is <i>report_name: Previewer</i> ; you can modify the title in Tools—>Tools Options
Viewing Region	The area where you browse through each page of your report, using buttons to navigate, split screens, or open a new Previewer window

Navigating Through the Previewer Window

You can navigate through the Previewer using the scrollbars and the iconic buttons in the toolbar at the top of the Previewer window. The toolbar includes buttons to perform the following actions:

Button	Description
First Page	Go to first page
Previous Page	Go to previous page
Next Page	Go to next page
Last Page	Go to last page
Page	Go to the page number that you enter in the field
Print	Print the report (to file or printer)
Mail	Send report to a MAPI-compliant mail system
Print Setup	Display the Print Setup dialog box, to change printer settings
Help	Display the Previewer page in the online Help system



1	New Previewer, Close Previewer options	3	Choose this rectangle to split Previewer horizontally
2	Zoom in, zoom out tools	4	Choose this rectangle to split Previewer vertically

Viewing Multiple Pages

To view the contents of more than one page at the same time, choose New Previewer to open another Previewer window. The window menu item shows the number of Previewer windows open at any time. Close each Previewer window individually.

Viewing Large Pages

When viewing large report pages, you can split the Previewer window either horizontally or vertically, to show the extreme left and right or top and bottom portions of a page.

- Horizontal split: Click and drag the gray rectangle above the vertical scrollbar.
- Vertical split: Click and drag the gray rectangle to the left of the horizontal scrollbar.

Release the mouse button at the position you want to split the viewing region.

Magnifying and Reducing the Report View

Zoom in and zoom out buttons are available on the toolbar. Select the large plus sign to zoom in and the small minus sign to zoom out.

Viewing Options

Screen and Preview use different fonts for formatting

- **Screen: screen fonts**
- **Preview: printer fonts**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE[®]

Viewing Options: Screen or Preview

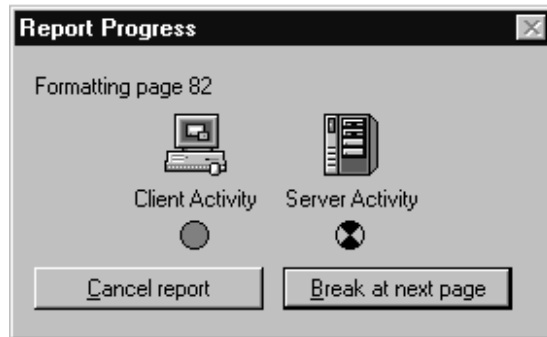
It is important to understand the difference between Preview and Screen. Both display the output in the Previewer window; however, they format the output using different fonts:

Destination Type	Description
Screen	Formats the output using the screen font that you specify in the Layout. It does not consider printer fonts.
Preview	Formats the output as if it were being sent to the printer. It uses printer fonts to format, but uses the screen fonts to display.

Using Preview gives you a better impression of how the printed result will appear. If users are likely to print reports after viewing in the Previewer, ensure that the destination is always Preview.

Canceling or Pausing Report Execution

Available in Microsoft Windows environment only



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Canceling or Pausing Report Execution

You can cancel or pause report execution while a report is running using the Report Progress dialog box. To halt execution of a running report:

- Select “Cancel report” to immediately cancel processing. You cannot continue formatting the report if you cancel processing, but you can still browse previously formatted pages.
- Select “Break at next page” to format to the end of the current page before canceling processing. This button appears in the dialog box only if you select (to format the last page) or type a page number (two more pages after the current page) in the Previewer window. You can browse previously formatted pages and then continue processing the report.

Note: This feature is only available in the Microsoft Windows environment.

Summary

- **Designing reports**
 - **Design structure**
 - **Database access**
 - **Common objects and code**
- **Executing reports**
 - **Output destinations**
 - **Screen or printer fonts**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Designing Reports

By discussing the report styles and structures, you should now be able to:

- Appreciate the principle of groups and frequencies of data
- Describe simple report styles and their underlying structure
- Realize the importance of database access and report performance, and consider this for every report you design
- Consider at the very early stages of development usage of common code and objects

Executing Reports

You have seen that there are a number of different ways to execute a report, all of which use the RWRUN60 executable.

- A report can have a run-time parameter form, allowing user input to control report data and output.
- Output can be to one of a number of different destinations, such as screen, file, or printer.
- If you choose a destination of Screen or Preview, you view the output in the Previewer window, where you can control many options including what page to view, magnify text, split screen, and the number of Previewer windows open.

Practice 2 Overview

- **Invoking the Reports Runtime executable**
- **Executing existing reports**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 2

This practice session contains:

- Invoking the Reports Runtime executable from the Program Manager
- Executing a number of different reports

Introduction

Before you learn to develop reports in the Report Builder, you should investigate the run-time executable that you use constantly in the production environment and for testing your reports prior to production.

This practice session consists of a combination of practical and paper-based questions.

Practice Session: Lesson 2

- 1 Invoke the Reports Runtime executable.
- 2 Run the report called `p2q2.rep`.
 - a Examine the report in the Previewer window. Among the styles discussed in this lesson as a reference, what style of report is it?
 - b How many groups of data are there in this report?
 - c In the Previewer window, use the scrollbars to move up and down the report to see all of the data.
 - d Close the previewer window. You are now back in the Runtime window.
- 3 From the Runtime window, run report `p2q3.rep`.
 - a In the Previewer window, move to the next page. Notice that this report has a header page with text. Notice that the Previewer also has many more buttons highlighted. This is because there are more available actions, such as Next Page, Previous Page, Last Page, and First Page.
 - b What style of report is this?
 - c In the Previewer window, experiment with the horizontal and vertical split screen features.
 - d Close the Previewer window. You are now back in the Runtime window.
- 4 From the Runtime window, run report `p2q4.rep`.
 - a What style of report is this?
 - b How many groups of data are in this report?
 - c How many pages are there in this report?
 - d Close the Previewer window. You are now back in the Runtime window.
- 5 From the Runtime window, run report `p2q5.rep`.
 - a What style of report is this?
 - b How many groups of data are in this report?
 - c Close the Previewer window. You are now back in the Runtime window.

-
- 6** From the Runtime window, run report `p2q6.rep`.
- a** Notice the Parameter Form. You are requested to input a customer ID. The report displays order information based on your input.
 - b** Specify a valid customer ID and run the report. Valid values include 201, 210, and 213; choose one of these and then choose Run Report.
 - c** What do you think would be a better method of providing valid values to a user?
 - d** Close the Previewer window. You are now back in the Runtime window.
 - e** Run the report again. This time do not specify a customer ID.
 - i** What is the result?
 - ii** In a production situation, what should happen in this instance?
 - f** Close the Previewer window. You are now back in the Runtime window.
 - g** Run the report again. This time enter Customer ID 999.
 - i** What is the result?
 - ii** In a production situation, what should happen in this instance?
 - h** Close the Previewer window. You are now back in the Runtime window.
- 7** From the Runtime window, run report `p2q7.rep`.
- a** In the Parameter Form notice the list of values for the Customer information. The report displays order information based on your input.
 - b** Select a valid customer name and run the report.
 - c** Close the Previewer window. You are now back in the Runtime window.
- 8** From the Runtime window, run report `p2q8.rep`.
- a** What do you notice that is different about this report?
 - b** Choose Details. Another report is called to display information about the orders the customer has placed with the company. This is called a Drill-Down report.
 - c** Close the Orders report Previewer window and then close the Customer report Previewer window. You are now back in the Runtime window.
 - d** Exit Reports Runtime.

Report Builder Concepts

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe the main Report executables**
- **Describe the main components of Report Builder**
- **Describe the main objects in a report**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

This lesson describes the Report executables and gives an overview of Report Builder, including a high-level description of its components and object hierarchy.

Objectives

At the end of this lesson, you should be able to do the following:

- Describe the main Report executables
- Describe the main components of Report Builder
- Describe the main objects in a report

Executables

- **Development and run-time**



- **Report background engine**



- **Remote report server**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report Executables

Introduction

Report executable filenames are operating-system specific.

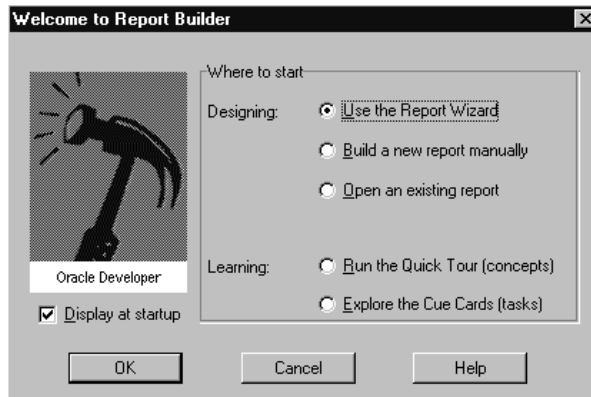
In Windows, all Reports executables follow a similar pattern: RW<?>60 . EXE .

For simplicity, the 32 suffix and .exe file extension are not included in the following table.

What Are the Main Reports Executables?

Windows Filename	Executable Title	Description
RWBLD60	Report Builder	Create, develop, and maintain report definitions.
RWRUN60	Report Runtime	Run-time environment for production or testing.
RWCON60	Report Converter	Converts a report definition to alternate storage formats.
RWMTS60	Report Server	Install/invoke a multithreaded remote server.
RWCLI60	Report Client	Send a report to a remote server.
RWRQM60	Report Queue Manager	View/schedule reports on a remote server.
RWRBE60	Report Background Engine	Invoke the local report background engine.
RWISV60	Interactive Server	Send a report to the local report background engine (Microsoft Windows only).

Invoking Report Builder



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Invoking Report Builder

In the previous lesson you used the run-time executable, `RWRUN60`, to run some reports. Now you invoke Report Builder, the `RWBLD60` executable, and look at the definitions of some reports, to describe the components and objects in the builder.

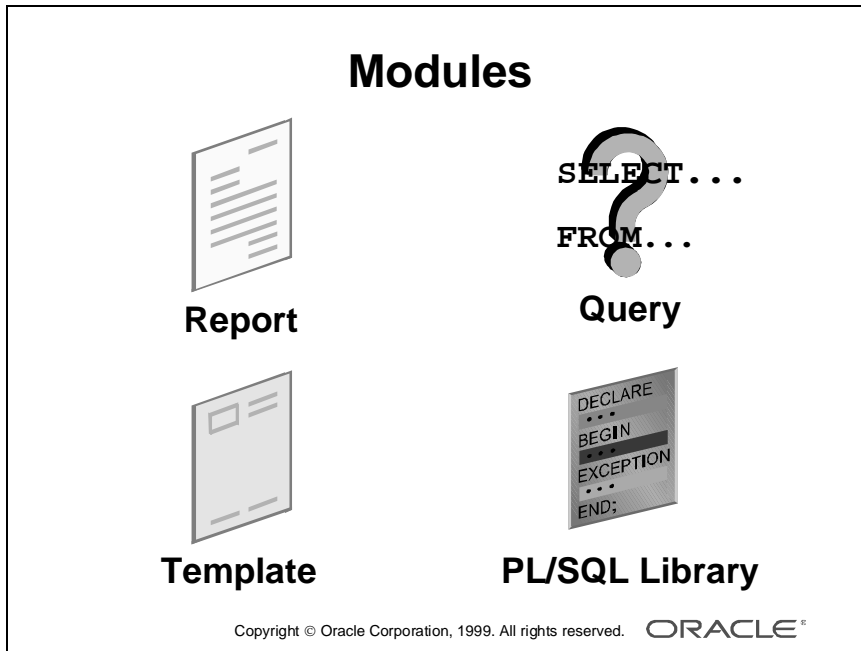
The Welcome Dialog Box

When you invoke Report Builder, the initial Welcome dialog box contains a check box “Display at start-up” that you clear if you want to suppress this screen.

If you clear this option, and then later want to see the Welcome dialog box, you must change the relevant option in the Preferences dialog box.

How to Display the Welcome Dialog Box

- 1 Choose Tools—>Preferences.
- 2 Select the Wizard tab.
- 3 Select the Welcome Dialog check box.



Report Builder Components

Introduction

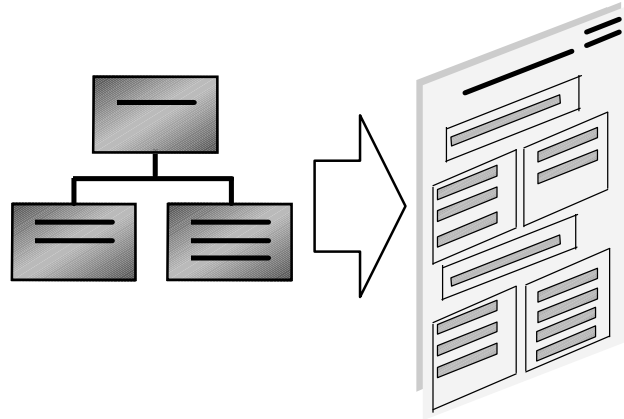
The Report Builder interface enables you to create a number of different types of modules, and it provides a Report editor in which you can view the structure and objects in a report module.

This section gives an overview of the module types and explains the different views in the Report editor.

Types of Report Builder Modules

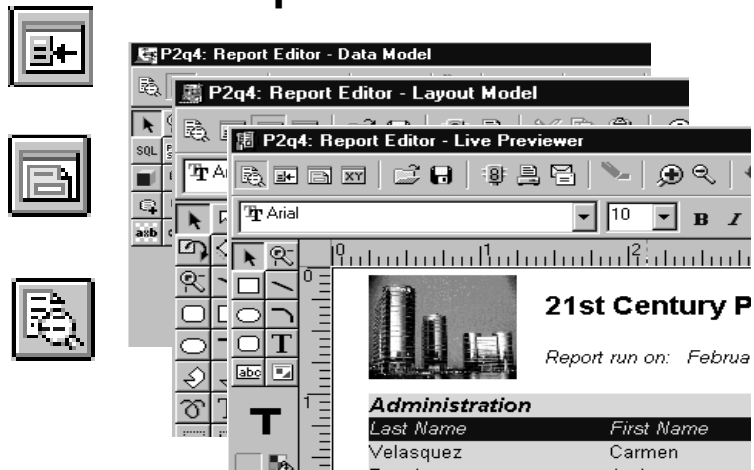
Module Type	Description
Report	A report definition
Query	An external query stored in a text file, or on the database, that you can reference from or copy into a report definition
Template	A skeleton definition containing common style and standards; can include graphics; provides a standard format to enable quick and easy development of professional standard look-and-feel reports
PL/SQL Library	A stand-alone library containing PL/SQL program units—procedures, functions, packages—that can be called from multiple Oracle Developer applications

Data and Layout



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report Editor



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report Data and Layout

An Oracle Developer report definition defines two main parts of a report and brings them together in the output.

- Data: Data structure and values to be displayed
- Layout: Formatting information about how the values appear in the output

The Report Editor

The Report editor contains two views to handle the data objects and layout objects separately:

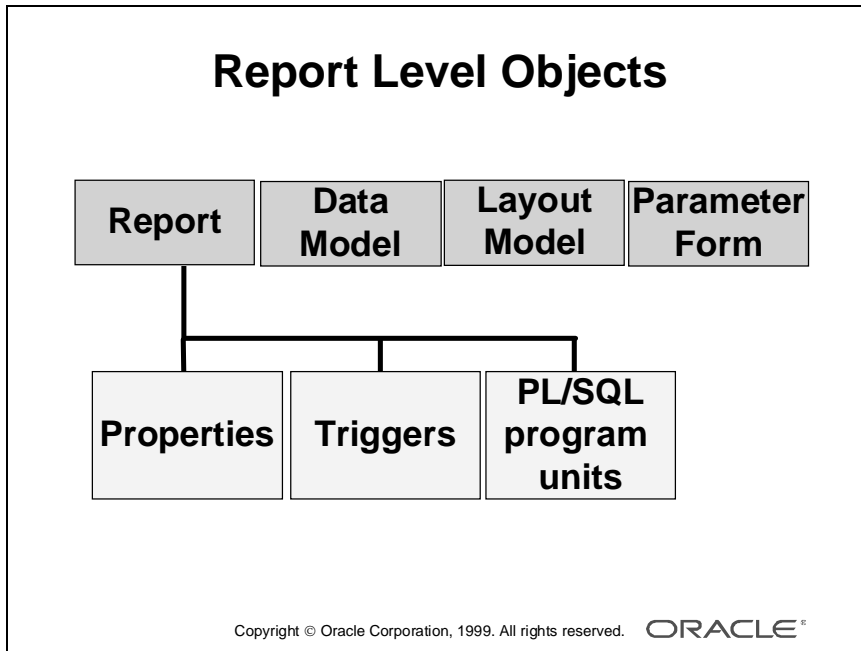
View	Description
Data Model	Displays a structural representation of the data in a report. The objects do not appear in the report output, but the structure determines the layout style, and the data objects provide the values that appear in the layout objects.
Layout Model	Displays the layout objects in a report and allows you to make many modifications to any layout object. All layout objects have properties that you can modify in a property palette. The hierarchy of the layout objects is determined by the Data Model.

The Report editor also contains the following two views:

View	Description
Live Previewer	Displays report output and allows you to make many commonly required, simple modifications to the layout, such as spacing, formatting fields, color, and editing text, without having to open the Layout Model.
Parameter Form	Displays the layout of the parameter form that, at run time, allows user-input of parameter values.

The Live Previewer displays the report output as it appears when printed. The other three views contain their own specific objects.

You can create many fully functional reports simply by using the wizard and modifying the report in the Live Previewer. However, this course also teaches you in later lessons how to use the Data Model, Layout Model, and Parameter Form so that you can create more complex reports if necessary.



Object Categories

Introduction

The Report Module consists of many objects that fall into the following four categories:

- Report level
- Data Model
- Layout Model
- Parameter Form

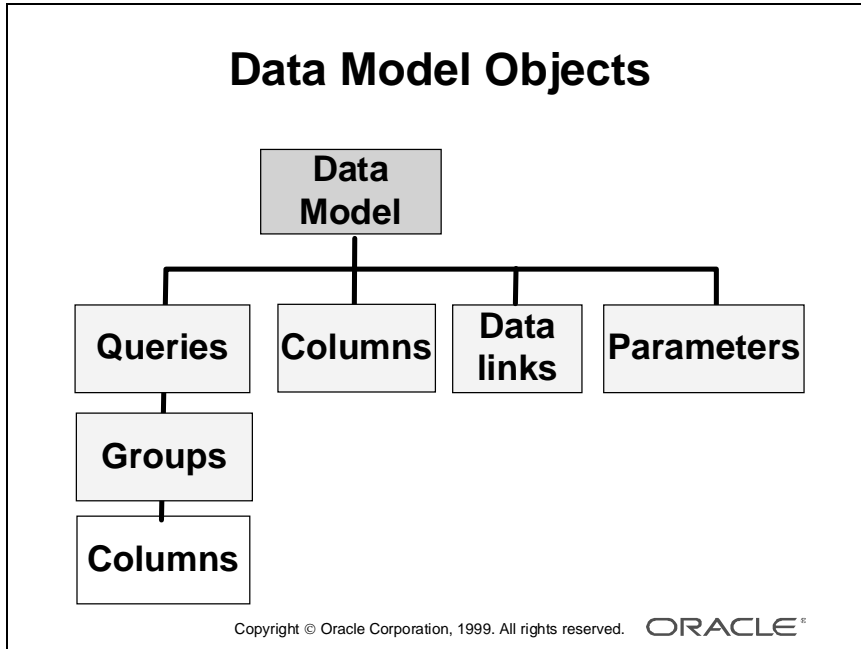
This section gives an overview of the objects in each category.

Note: All the objects mentioned in this section are discussed in greater detail in later lessons.

Report-Level Objects

The report objects define the measurements, dimensions, triggers, and PL/SQL program units of a report. The report object itself consists of the following:

Object	Description
Properties	Define page dimensions and Previewer settings, for example
Triggers	Allow PL/SQL to be executed at different stages of the report execution
PL/SQL Program Units	Contain functions and procedures that can be called from report-level objects in the same report



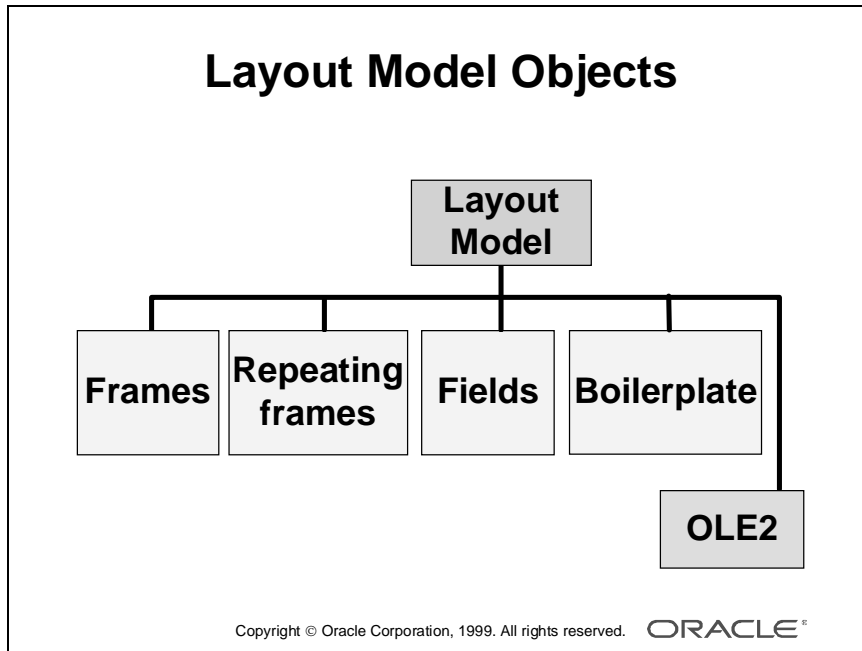
Data Model Objects

The Data Model objects for a report define the data used in the report and the data structure. Data Model objects appear in the Data Model editor and the Object Navigator. Data Model objects can be of the following types:

Object	Description
Query	Select the data for your report
Group	Organize the data to form the required hierarchical structure
Columns	Contain individual data values; database columns exist by default and contain data from the database columns or expressions defined in the query; you can also create Formula, Summary, and Placeholder column types
Data Link	Join queries for complex data relationships
Parameter	Provide for run-time defaults or user input; system parameters exist by default; you can also create User Parameters

With the exception of parameters, you create all objects in the relevant editor, not in the Object Navigator.

Parameters do not appear in the editor. You create and modify parameters in the Object Navigator only.



Layout Objects

The Layout Model objects define the format of the report including the positioning and appearance of data, text, and graphics. The main layout objects fall into the following categories:

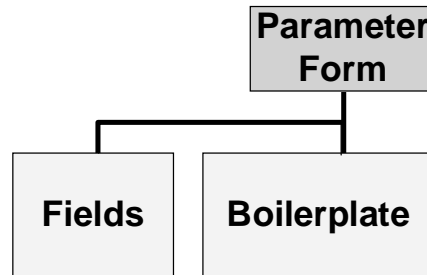
Object	Description
Repeating frame	Contains other objects and prints once for each record of the associated group
Frame	Contains other objects and prints only once
Field	Contains data and other variable values and their formats
Boilerplate	Contains text or graphics that may appear anywhere in the report

These and other layout objects are discussed later in the course.

OLE2 Support

Report Builder supports the use of OLE2. The Layout Model includes an OLE2 tool that allows you to create an OLE2 container and define the contents.

Parameter Form Objects



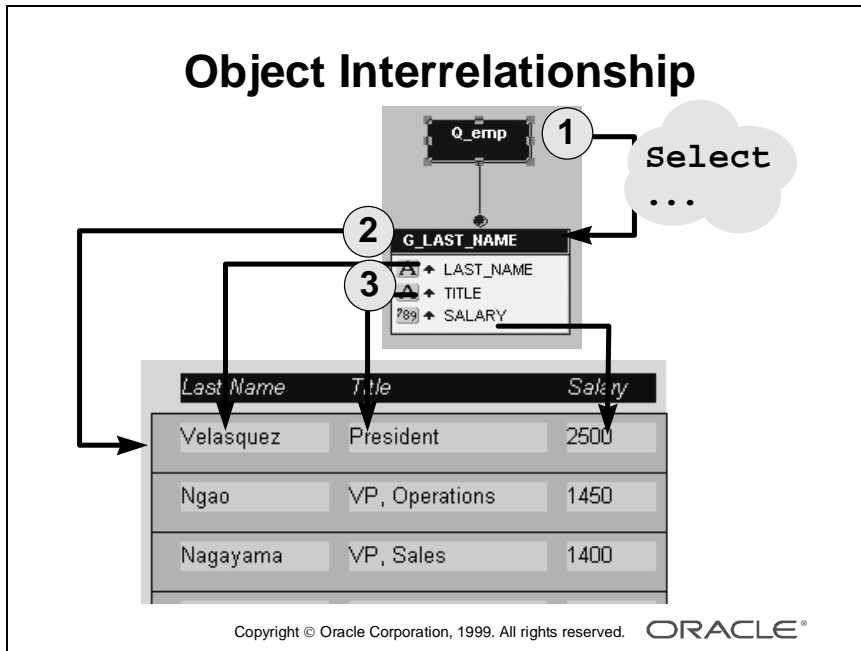
Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Parameter Form Objects

The Parameter Form objects define the appearance of the run-time parameter form. You create and modify run-time parameter form objects.

Object	Description
Field	Contains parameter values
Boilerplate	Contains constant text or graphics that appear on the run-time parameter form

Note: The Parameter Form controls the layout of the run-time parameter form. The objects are similar to layout objects. The source of a parameter field comes from a parameter that is a Data Model object. Parameters appear in the Object Navigator, not in the Data Model view.



1	Each query fetches data records and structures them in the group hierarchy.
2	Each group is the source of a repeating frame.
3	Each column is a source of a field.

Object Relationships

Now that you know the different categories of objects, it is also important to understand the relationship between these objects. The diagram opposite shows the relationships between some of the most common objects, explained in terms of a simple tabular report.

Data Model Objects

- A *query* fetches records from the database.
- Each fetched record becomes an instance of the related *group*.
- Each database value is fetched into the related *column*.

Layout Objects

- A *column* provides the value that is displayed in one or more layout *fields*.
- A *field* must display all instances of its related *column* value; therefore, each record instance of a *group* is represented by a *repeating frame* that contains all the relevant fields.

Summary

- **Executables for development and deployment**
- **Report Builder modules:**
 - Report, Template, External Query, PL/SQL Library**
- **Report editor: switch views**
 - Live Previewer, Data Model, Layout Model, Parameter Form**
- **Object categories:**
 - Report, Data, Layout, Parameter**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Oracle Developer provides a number of executables for developing and deploying reports, including Report Builder.

The Report Builder interface allows you to develop four types of modules, including report definitions.

Report Builder provides a Report editor in which you can view and modify the objects that the wizard creates. You can also create your own objects to enhance your report structure and layout.

The Report editor allows you to switch views, depending on the objects that you want to modify.

Objects in a Report module fall into four categories: Report, Data Model, Layout Model, and Parameter Form.

Practice 3 Overview

- **Invoking Report Builder**
- **Opening an existing report**
- **Switching views in the Report editor**
- **Accessing the Help system**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**[®]

Practice Session Overview: Lesson 3

This practice session contains:

- Invoking Report Builder
- Opening an existing report
- Switching views in the Report editor
- Accessing the Help system

Introduction

The questions in the practice session provide an introduction to the Report Builder interface. You open and navigate through an existing report definition and use the Help system to obtain information about some of the Report executables.

Creating Reports Using the Report Wizard

Objectives

At the end of this lesson, you should be able to do the following:

- **Create a simple tabular report using the Report Wizard**
- **Describe the different methods of building the report query**
- **Summarize report values**
- **Modify the style and content of a report**
- **Create other report styles available in the wizard**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

In this lesson you learn how to create a simple tabular report and modify the report by adding data and altering the report style. This lesson also shows you how to create other styles of report that are available from the Report wizard.

Objectives

At the end of this lesson, you should be able to:

- Create a simple tabular report using the Report wizard
- Describe the different methods of building the report query
- Summarize report values
- Modify the style and content of a report by reentering the wizard
- Create other report styles available in the wizard

Invoking Report Wizard



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Invoking Report Builder and the Report Wizard

Introduction

Remember, from a previous lesson, that you can invoke Report Builder either directly from the Oracle Developer menu or from within Project Builder.

In both cases, when you open Report Builder, the Welcome screen gives you the option of using the wizard to build a new report.

The Report Wizard

The Report wizard provides an easy step-by-step interface to create a new report.

The Report wizard has a Welcome page. To suppress this page, clear the “Display at startup” check box.

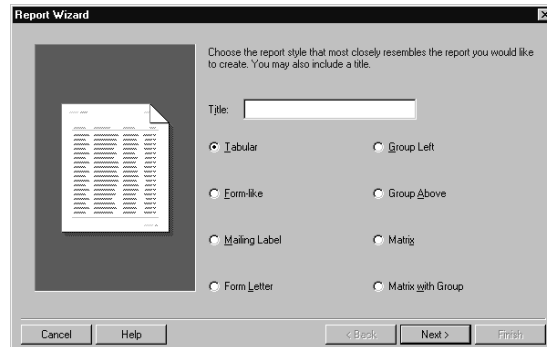
You can reinstate this page in the same way as the Welcome dialog box in Report Builder; select the Wizard tab in the Preferences dialog box and then select Report Wizard Welcome Page.

Each page of the Report wizard asks you for information to help you create your initial report. Step through the wizard pages, choosing Next and Back, until you are satisfied with the initial information that you have entered. On the last page, choose Finish.

Creating a Tabular Report

Wizard Pages

- Report Style
- Query Type
- Data Query
- Fields
- Totals
- Labels
- Template



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating a Tabular Report Using the Wizard

Introduction

You can use the wizard to build eight styles of reports. In this section, you step through the wizard pages to create a tabular report. Later in this lesson, you create some other report styles.

Report Styles

The first wizard page, after the Wizard Welcome, shows the various styles of report. Select Tabular and then choose Next.

Wizard Pages

When you choose the Tabular style of report, the wizard steps you through the following pages:

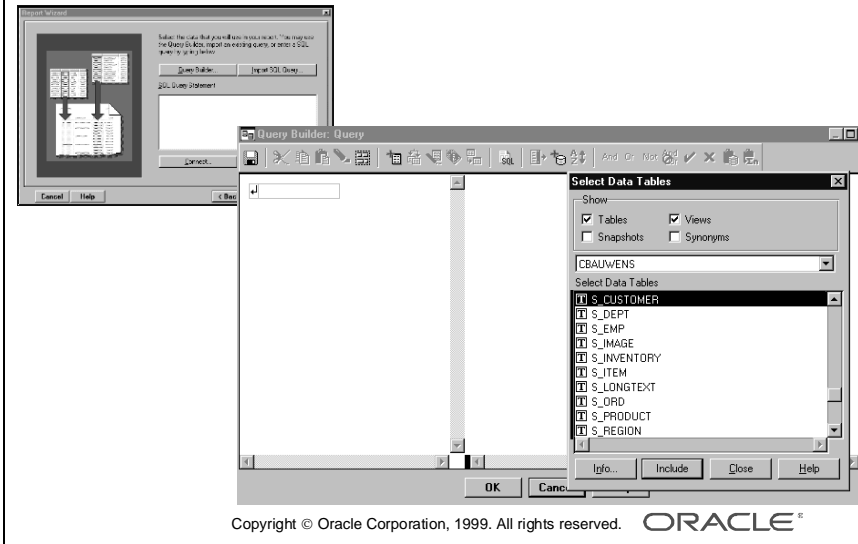
Page Name	Description
Type	Select the query type on which you want to base your report.
Data	Enter a SELECT statement to retrieve the report data.
Fields	Select the fields that you want to display in the output.
Totals	Select the fields that you want to summarize.
Labels	Alter the labels that appear for each field and the width of each field.
Template	Select the template that you want to use for this report. A template contains standard information such as company logo, date, and so on.

Query Types

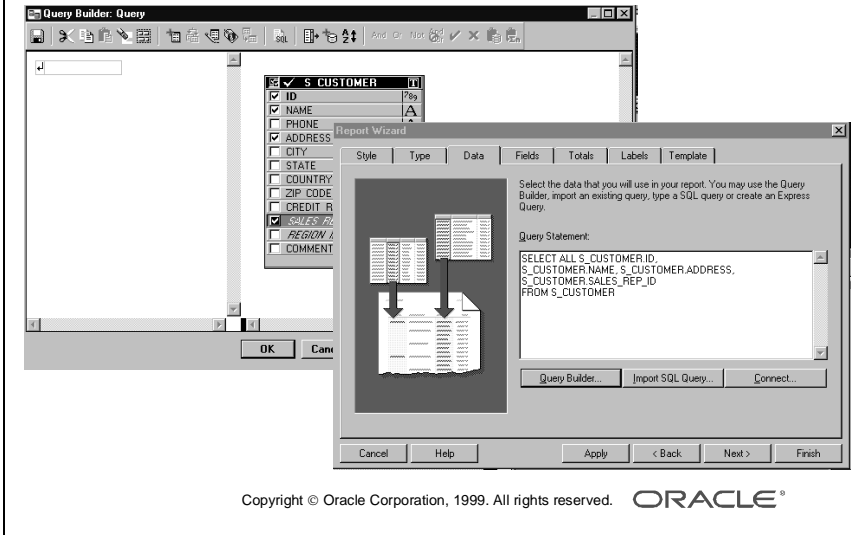
On the second page, the Report Wizard enables you to choose whether to build a SQL query or an Express query.

Note: Express delivers online analytical processing (OLAP) with capabilities that include forecasting, what-if scenarios, and financial modeling. Based on a multidimensional data model, Express is optimized for the query and analysis of corporate data such as sales, marketing, financial, manufacturing, and human resource data.

Using the Query Builder



Building a Query



Using the Query Builder

Build your query using the Query Builder GUI. This saves you time and increases the ease-of-use for developers not familiar with building SQL statements or with the application tables.

How to Use the Query Builder

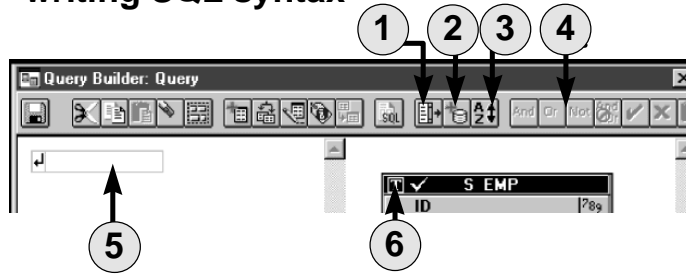
- 1 Choose Query Builder from the query page in the Report wizard.
- 2 Enter your username, password, and alias in the Connect dialog box that appears if you have not already connected to the database.
- 3 Select the data tables to build the query.
- 4 Choose Include. The tables appear in the selection area.
- 5 Choose Close to close the Select Data Tables window.
- 6 In each table, double-click the column names that you want in the query, or use the check box. To select all columns, double-click the Table title bar.
- 7 Choose OK.

Query Builder copies the query syntax into the Report wizard. You can modify the query by reentering the Query Builder or by modifying the SQL query statement text.

Note: If you prefer to write your own SQL statement, enter the syntax directly in the SQL query statement area of the Query page. Alternatively, you can import the contents of a file by choosing Import SQL Query.

Query Builder Functions

- Query Builder functionality embedded in Report Builder
- Easy User Interface: alternative to writing SQL syntax



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Column sequence	4	Logical operators for conditions
2	Define column	5	Conditions box for WHERE and HAVING clauses
3	Sort	6	Object type (T=Table, V=View; S=Synonym; A=Alias)

Query Builder Functions

This course does not teach the detailed functionality of Query Builder. You can find comprehensive help in the Report Builder online Help system: [Help](#)—>

[Contents](#)—>[Building and Running a Report](#)—>[Defining a Query](#)—>[Using Query Builder](#).

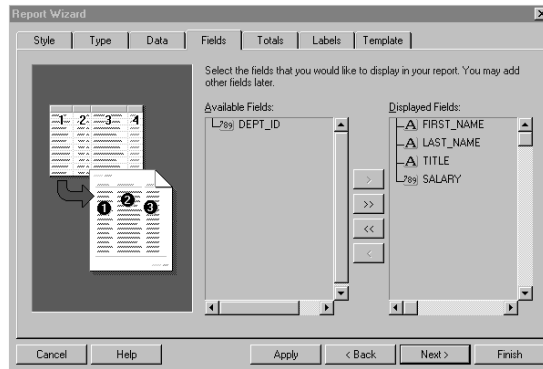
You can use Query Builder to build almost any query that you can write as a SQL SELECT statement.

You can select from a table, view, or synonym. The letter T, V, or S appears in the object title bar to indicate which it is. If you define the same table more than once, Query Builder creates an alias (A).

Following is a brief description of some Query Builder toolbar buttons and functions.

Function	Description
Column Sequence	Defines the sequence of column names in the SELECT clause
Define Column	Defines a new, derived, or calculated column to be included in the SELECT clause; use this definition to calculate and retrieve derived values from the server
Sort	Defines the ORDER BY clause; select the columns that you want to be sorted, and choose Ascending or Descending sorting order
WHERE and HAVING clauses	Place the cursor in the Conditions box on the left side of the Query Builder window. Type a condition. Use the AND, OR, and NOT buttons to create compound conditions.

Choosing Displayed Fields



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Choosing Displayed Fields

In the Field page, select each field from the list of Available Fields and choose >. The selected fields move to the Displayed Fields list.

To display all fields, choose >>.

Sequencing Displayed Fields

You can alter the sequence of displayed fields by dragging one field above or below another in the list. The sequence of fields in this list determines how the fields appear in the report output. In a tabular report, the fields appear in sequence from left to right across the report.

Referencing Nondisplayed Fields

Fields that remain in the Available Fields list are available for you to reference in your report definition as hidden fields or in PL/SQL trigger code.

In the report output, the user sees only those fields that you transfer to the Displayed Fields list.

Summaries and Labels

The screenshot displays two overlapping windows from the Oracle Report Wizard. The primary window is the 'Totals' tab, which allows users to select fields for calculation and choose the type of summary function. The 'Available Fields' list includes LAST_NAME, FIRST_NAME, TITLE, SALARY, and DEPT_ID. The 'Totals' list shows 'Sum(SALARY)'. Summary functions available include Sum, Average, Count, Minimum, Maximum, and % Total. The secondary window is the 'Labels' tab, which is used to define the labels and widths for the report fields. It contains a table with the following data:

Fields and Totals	Labels	Width
LAST_NAME	Last Name	10
FIRST_NAME	First Name	10
TITLE	Title	10
SALARY	Salary	9
Sum(SALARY)PerReport	Total	9

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summaries and Labels

The next two pages of the Report wizard allow you to create summaries based on any of the displayed fields and to modify the labels and width of the displayed fields.

Summaries If you choose a summary, the wizard creates the summary at each level of the report; that is, at report level and also at each break level, if your report contains break groups.

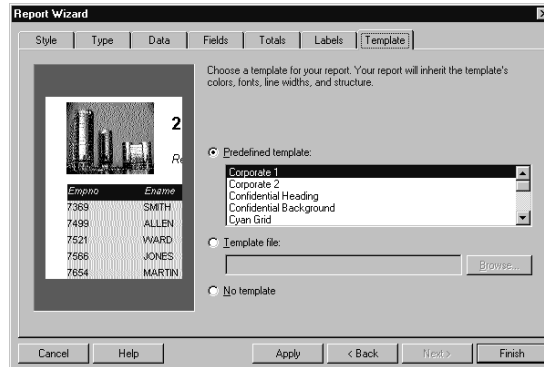
You can deselect some of these summaries later (by reentering the wizard) if, for example, you want a report-level summary only.

Labels The field label is displayed on one or more lines in the report output. In a tabular report, the labels appear above the field values.

- If the initial label is wider than the field, Report Builder allows enough space for the label, or displays it on multiple lines.
- If you increase the number of characters in the label text in the reentrant wizard, the label may appear truncated in the report output.

Choosing a Report Template

- Enforce corporate standards
- Create professional-looking reports easily



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Choosing a Report Template

Choose a template from the list of predefined template names.

In a template, fonts, styles, and colors are already selected for designated objects.

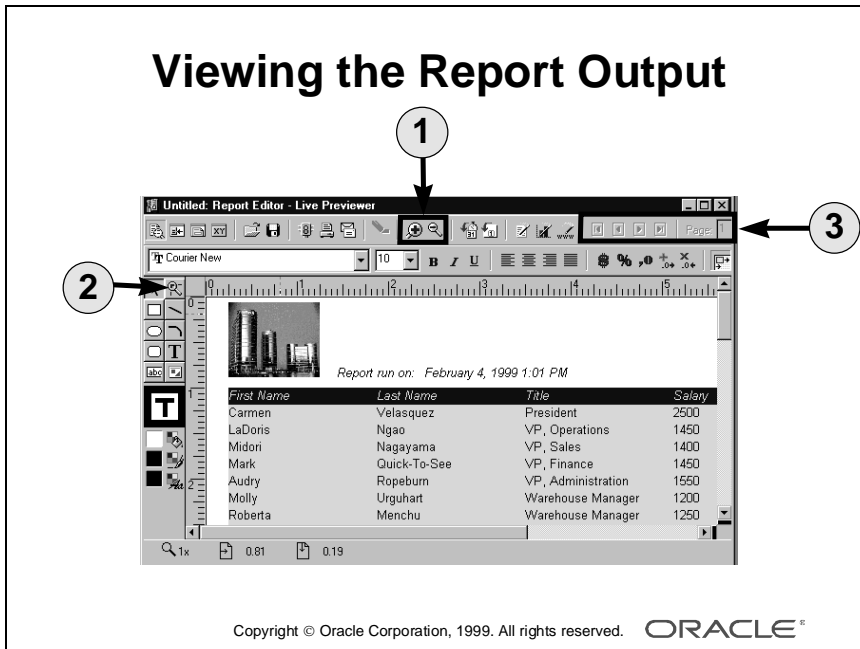
A variety of templates are available with the standard Reports installation.

How to Select a Predefined Template

- 1** Select the Predefined Template option button, if it is not already selected.
- 2** Select a template from the Template list.
- 3** Choose Finish.

Note: How to modify and use your own user-defined templates is covered later in the course.

Viewing the Report Output



1	Zoom buttons
2	Magnify tool
3	Page buttons

Viewing the Report Output

When you finish creating your report in the Report wizard, the output appears in the Live Previewer.

Magnifying the Output

Like the run-time previewer, the Live Previewer toolbar contains Zoom In and Zoom Out buttons to allow you to magnify or reduce the size of the output.

The Live Previewer also contains a Magnify tool in the vertical toolbar. This provides a view of the area of layout you want to see.

- To increase the view size, select the Magnify tool and click in the layout area.
- To reduce the view size, select the Magnify tool, hold down the Shift key, and click in the layout area.

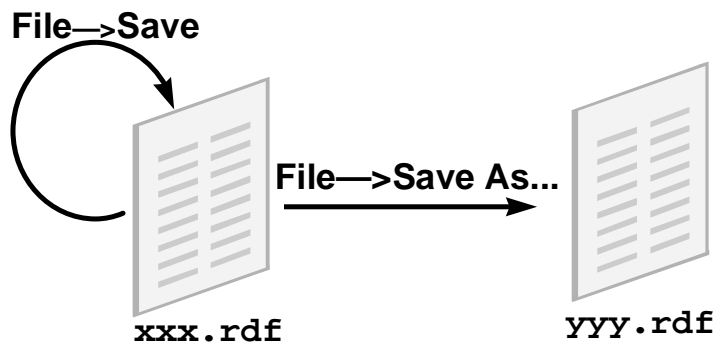
Viewing Different Pages

Like the run-time previewer, the Live Previewer toolbar contains four buttons, and the specific page option, to allow you to scroll through the pages of your report.

Other functions of the Live Previewer are discussed in the next lesson.

Saving the Report Definition

Save changes frequently!



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Saving the Report Definition

Remember to save the report frequently by choosing Save in the toolbar, or the File—>Save menu option.

The file is saved with an `.rdf` extension. Storage methods are discussed later in the course.

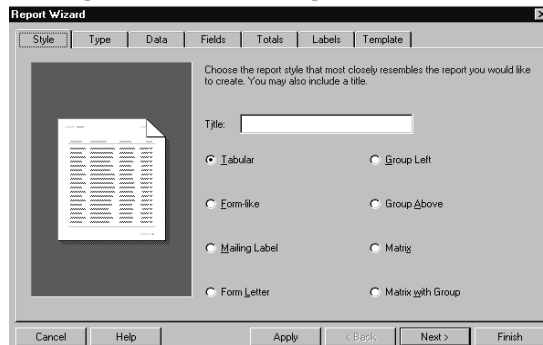
Copying the Report Definition

If you want to make a copy of the report definition in a different filename, use the menu option File—>Save as.

There is no toolbar button for the Save as option.

Reentering the Wizard

- Choose Tools—>Report Wizard
- Tabs are different for each report style
- Wizard preserves all previous settings



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Other Report Styles

Introduction

In this section you discover the other report styles available in the wizard and the pages that the wizard displays. There are two methods of creating additional report definitions with the Report wizard:

- Modifying an existing report definition by reentering the wizard
- Creating a new report definition by restarting the step-by-step wizard

Modifying a Report by Reentering the Wizard

The reentrant Report wizard preserves your current settings and query. You can make changes to these or other settings and choose Finish when you are ready to reapply all the wizard settings to your report.

To reenter the wizard in an existing report definition, follow these steps:

- 1 Choose Report Wizard from the horizontal toolbar of the Report editor.
- 2 Choose Tools—>Report Wizard.
- 3 In the Object Navigator, choose Report Wizard from the right-mouse-button menu.

Tabbed Pages

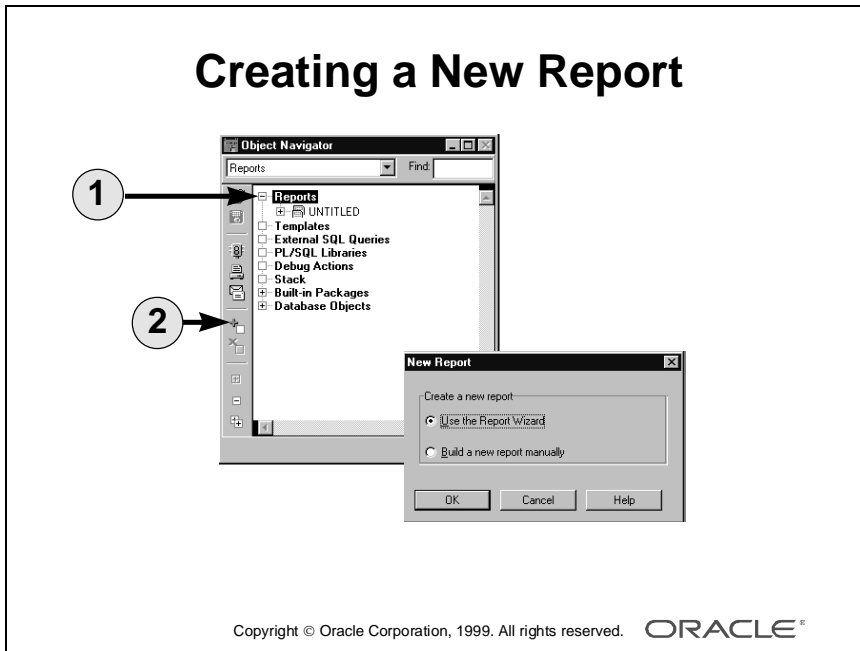
When you reenter the wizard, you see a tab for each page. You can navigate directly to the page you want to modify instead of choosing Next to step through each page in turn.

Choose Finish at any time to apply the changes.

Creating a Form-Like Report

There are three significant features in the Form-like style that differ from the Tabular style.

- Labels appear to the left of each field.
- Each field appears to the left of the previous field, across the page.
- Each record appears on a new page.



1	Report node
2	Create object button

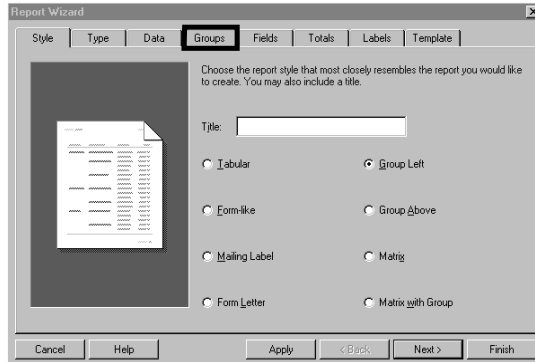
How to Create a New Report

To create a new report with the wizard when you are already in the Report Builder, follow these steps:

- 1** Select the Report node in the Object Navigator.
- 2** Choose Create, the green plus sign in the vertical toolbar.
- 3** Choose Use the Report Wizard.

Creating Break Reports

- Break report styles:
Group Left, Group Above
- Additional wizard page:
Groups



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Break Report Labels

Group Left

Region	Department	Last Name	Title
xxxxxx	xxxxxxx	xxxxxxx	xxxxxx
		xxxxxxxx	xxx

Group Above

Region	xxxxxxxxx
Department	xxxxxx
<u>Last Name</u>	<u>Title</u>
xxxx	xxxxxxxxxx
xxxxxxxxxx	xxxx

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Break Reports

When you select one of the break (or master-detail) styles—Group Left or Group Above—the wizard displays an extra page, called the groups page, in which you choose the columns that should be in each break group of the report. You can choose multiple levels of breaks.

Group Left The output is displayed as columns across the page, with the groups next to each other.

All labels appear above the fields.

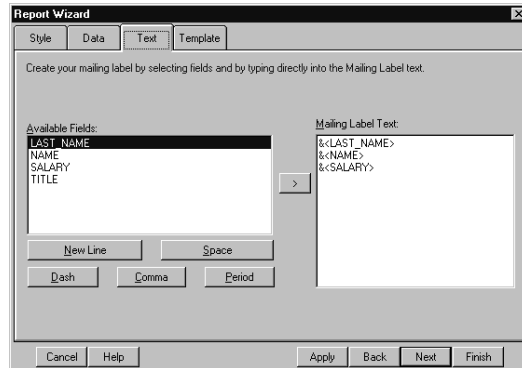
Group Above The output is displayed with the groups below each other nested within the parent group. The labels for all master groups appear to the side of the fields.

Labels for the bottom detail groups appear above the fields, as in a tabular report.

Creating Mailing Labels and Letters

Wizard pages

- Style
- Data
- Text
- Template



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Mailing Labels and Letters

When you choose the Mailing Label or Form Letter report style, the wizard displays only four tab pages: Style, Data, and Template pages as before, and the Text page.

How to Create Text

- 1 Select a field from the Available Field list.
- 2 Choose > to display in the Text area.
- 3 Type new lines and punctuation as required, or use the punctuation buttons supplied: New Line, Space, Dash, Comma, and Period.
- 4 Select additional fields from the Available Field list.

What Appears in the Text Area?

The selected field name appears in the text area, enclosed in optional angled brackets (<>) and prefixed by an ampersand (&). This indicates that the field name is a variable. Each variable is replaced by a specific value at run time.

You can type directly into the text area instead of selecting from the Available Fields list. However, remember to prefix each field name (variable) with an ampersand.

Any word that you type without an ampersand appears as a text string in your mailing label output. For example:

Text Area	Output
LAST_NAME &<LAST_NAME>	LAST_NAME Velasquez LAST_NAME Ngao

The optional angled brackets allow you to display two variables side by side with no separating space.

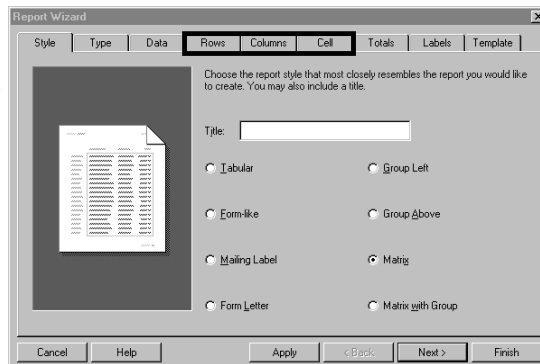
What Is the Difference Between Mailing Labels and Form Letters?

Mailing Label	Form Letter
Multiple records on one page	One record on each page

Creating a Matrix Report

Three additional Wizard pages

- **Matrix rows**
- **Matrix columns**
- **Matrix cells**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Matrix Reports

When you choose the Matrix report style, the wizard displays three new tab pages.

Tab Page	Description
Rows	The field to be displayed vertically down the left side of the matrix; you can choose multiple levels of rows to create a vertically nested matrix
Columns	The field values to be displayed horizontally across the top of the matrix; you can choose multiple levels of columns to create a horizontally nested matrix
Cell	The field value that becomes the cell, or cross-product, of the matrix

Creating Matrix Summaries

When you choose a summary in the Summary page, the wizard creates three summaries in the matrix.

Summary	Description	Position in Output
Row	One value for each row	Right side of matrix, at end of row
Column	One value for each column	Bottom of matrix, below column
Report	One value for the report	Bottom right corner of matrix

Creating a Matrix for Each Group Record

Choose the Matrix with Group report style. This provides a similar group structure to the Group Above report.

In your Regional report, you can select REG_NAME in the Group page so that the wizard structures the Department/Title matrix for each record.

The differences from a nested matrix are:

- Only the relevant Departments and Titles are displayed for each region.
- If you create summaries for a Matrix with Group style, the wizard calculates and displays summaries for each group as well as the report total. The report total is displayed at the end of the report, in the bottom left corner.

Summary

- **Rapidly create reports with the Report Wizard.**
- **Build queries using the Query Builder.**
- **Apply templates to reports.**
- **Modify reports by reentering the wizard.**
- **Create new reports of different styles.**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Report Styles

There are eight common styles of report that you can create by using the Report wizard. The wizard steps you through a variety of pages, depending on your chosen report style.

Query Builder

The built-in Query Builder provides a user-friendly graphical interface for you to build the SQL queries to retrieve your report data. Alternatively, you can write the SQL statement yourself, or import the contents of a file.

Predefined Templates

Predefined templates offer standard fonts, colors, formats, and images that enable you to create many reports with a professional appearance and a standard look and feel very quickly and easily.

Reentrant Wizard

The wizard is reenterable; it retains all your previous setting, and allows you to modify any settings by selecting the relevant tabbed page.

Practice 4 Overview

- **Creating a Tabular report**
- **Modifying the report, to create a Group Above break report**
- **Creating a Form Letter**
- **Creating a Matrix report**
- **Creating a Matrix with Group report**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 4

This practice session contains:

- Creating and saving a Tabular report
- Modifying the report to create and save a Group Above break report
- Creating and saving a Form Letter
- Creating and saving a Matrix report
- Creating and saving a Matrix with Group report

Introduction

The questions in the practice session are intended to ensure that you have a good understanding of how to create various styles of report using the Report wizard.

Note: When you are completing the practice sessions, use the file-naming conventions that the questions specify. You may need these files in later practices.

In addition, note the field headings and widths. With many reports, you need to change these to achieve the desired result.

Many of the practice sessions ask you to rename files. Be sure to use Save As, not Save, so that you do not overwrite the existing filename.

Practice Session: Lesson 4

How to Invoke Report Builder and the Wizard

If Report Builder is not open, launch it from Project Builder and create a new report using the Report wizard.

If Report Builder is open after the previous practice session:

- Close any reports currently in the Object Navigator.
- Create a new report by selecting the Report node and choosing Create Object.

Note: There is an alternative method: If no reports are open in the Object Navigator, you can double-click the Report node.

- 1** Create a tabular report containing the following data. Use Query Builder to build the query.

```
S_DEPT: NAME
```

```
S_EMP: FIRST_NAME, LAST_NAME, TITLE, SALARY, COMMISSION_PCT
```

Display all fields. Do not make any changes to the other wizard pages.

- 2** Using the reentrant Report wizard, make the following modifications; testing your report frequently:
 - a** Modify the query to sort the data in descending order of salary.
 - b** Add a summary to show the total salary value.
 - c** Reduce the width of SALARY and TOTAL to 6.
 - d** Change the COMMISSION_PCT label to Comm and reduce the width to 4.
 - e** Save the report to a file named p4q2.rdf.
- 3** Using the reentrant wizard, modify the report to create a break report.
 - a** Choose report style Group Above.
 - b** Group the report by the department name.
 - c** Choose the template called Bright 1.
 - d** Save the report as p4q3.rdf.

-
- 4** Create a new report containing the following data.
- a** Use the Report Wizard.
 - b** Choose Mailing Labels style.
 - c** Use Query Builder to build the query.


```
S_CUSTOMER: NAME, ADDRESS, CITY, STATE
S_ORD: ID, DATE_SHIPPED, TOTAL
```
 - d** In the text area, include all four customer fields, with each field on a new line.
 - e** Select the template Draft and choose Finish.
 - f** Save the report to a file named `p4q4.rdf`.
- 5** Using the reentrant wizard, modify the report to create a Form Letter style.
- a** In the text area, enter some free-flowing text for each order. For example:


```
Order No. &<ID> was shipped to &<NAME> in &<CITY> on
&<DATE_SHIPPED>.
```
 - b** The order has a total value of `&<TOTAL>` dollars.
 - c** Select a different template and choose Finish.
 - d** Save the report as `p4q5.rdf`.
- 6** Create a new report.
- a** Choose Matrix Report style.
 - b** For the query, import the contents of `p4q6.sql`.
 - c** Display customer names down the left side of the page.
 - d** Display product numbers across the top of the page.
 - e** Display the sum of total value in the cells.
 - f** Create a summary to give the total of the `sum(total_value)` values.
 - g** Reduce the widths of `PRODUCT_ID` to 5 and all four summaries to 7.
 - h** Change the `PRODUCT_ID` label to Product.
 - i** Choose a template that gives Landscape output.
 - j** Save the report to a file named `p4q6.rdf`.

If You Have Time

- 7** Modify the matrix report to create a Matrix with Group.
- a** Choose Month as the group and check that all totals have a width of 7.
 - b** Save the report as `p4q7.rdf`.

Note: The query for the matrix reports above has been restricted to display only products beginning with 4, so that you can see and understand the complete matrix more easily.

Enhancing Reports in the Live Previewer

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe the report Live Previewer**
- **Modify the display of report data in the Live Previewer**
- **Modify the positioning of report data**
- **Highlight data using conditional formatting**
- **Add page numbering and current date**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

In this lesson you learn to enhance your reports with the most commonly required changes, while viewing the live run-time output in the new Live Previewer in Report Builder.

Objectives

At the end of this lesson, you should be able to do the following:

- Describe the report Live Previewer
- Modify the display of report data in the Live Previewer
- Modify the positioning of report data in the Live Previewer
- Highlight data using conditional formatting
- Add page numbering and the current date to a report.

What Is the Live Previewer?

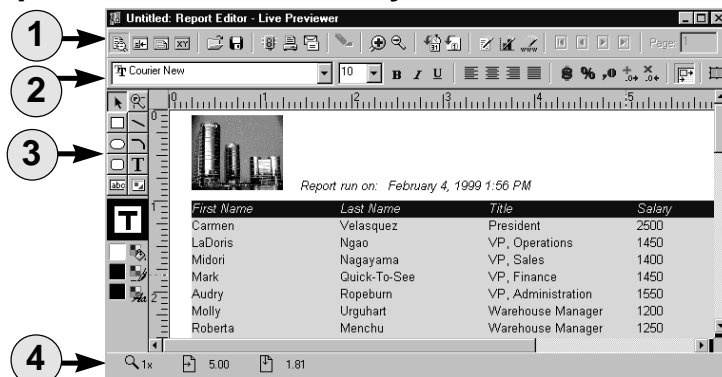
A view of report output that allows live editing of text and layout attributes

- True WYSIWYG report editing
- Easy editing: See it! Click it! Change it!
- Cached report data

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

The Live Previewer Window

Save report with open Live Previewer:
report runs automatically next time.



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Toolbar	3	Tool palette
2	Stylebar	4	Status bar

What Is the Live Previewer?

Introduction

The report Live Previewer is a WYSIWYG editor. All edits that you perform in the Live Previewer are on live data.

Cached Report Data

Oracle Developer caches the data the first time that you run the report, then reuses the cached data each time you run the report during the session, unless you modify the report in any way that requires refreshed data, e.g. changing the group structure or adding a summary column.

The Live Previewer Window

The Live Previewer has a horizontal toolbar and stylebar that contain common functions, also available from the pull-down menu.

The vertical tool palette provides tools that allow you to create simple boilerplate objects and text to enhance your report, as well as color palettes to modify the color fill and borders of objects.

You can suppress the tool palette and status bar from the pull-down menu.

Saving Reports

Save your report definition often, using the Save icon on the toolbar or the Save or Save As option on the File menu, to preserve the changes that you make in the Live Previewer.

When you open an existing report, it retains its last saved state. If you save a report while the Live Previewer is active, it automatically runs and displays the current data when you next open the report.

If you do not want the report to run at the time you open it, close the Report editor, or change the view to Data Model or Layout Model before saving the report definition.

Modifying a Report

Common modifications

- **Align columns**
- **Set format masks**
- **Manipulate objects**
- **Edit text**
- **Modify visual attributes**
- **Highlight data using conditional formats**
- **Insert page numbers and current date**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying a Report in the Live Previewer

Introduction

This section discusses some of the most commonly required changes that you need to make to reports, after creating the initial definition with the wizard:

- Aligning columns
- Setting format masks
- Manipulating objects
- Editing text
- Modifying visual attributes
- Highlighting data using conditional formatting
- Inserting page numbering
- Inserting current date and time

Aligning Columns



Last Name	Salary
Velasquez	2500
Ngao	1450
Nagayama	1400
Quick-To-See	1450
Ropeburn	1550
Urguhart	1200
Menchu	1250
Biri	1100
Catchpole	1300
Havel	1307
Magee	1400
Gijum	1490
Sedeghi	1515

**Point and
select**

Last Name	Salary
Velasquez	2500
Ngao	1450
Nagayama	1400
Quick-To-See	1450
Ropeburn	1550
Urguhart	1200
Menchu	1250
Biri	1100
Catchpole	1300
Havel	1307
Magee	1400
Gijum	1490
Sedeghi	1515

**Choose
“end justify”**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Aligning Columns

In the sample report output, in the left panel, the salary values are aligned to the left. You often need to align numbers to the right.

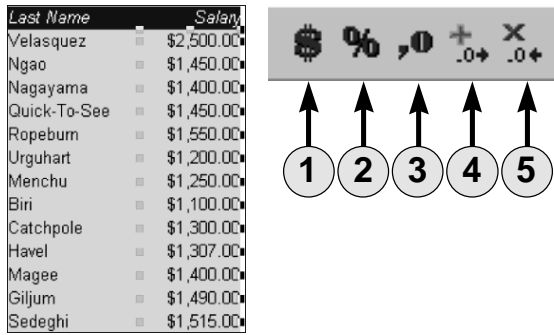
- 1 Select the column that you want to align.
- 2 Choose the “end justify” icon on the stylebar.

Aligning Multiple Objects

You can align each object separately, so that you can center the labels if you want. However, if you want to align the values and the labels to the right, use shift-click to select more than one object at one time.

Setting a Format Mask

Change format masks from the stylebar



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Currency symbol	4	Add decimal places
2	Percentage symbol	5	Remove decimal places
3	Thousand separator		

Setting a Format Mask

The most commonly used numeric format symbols are available on the stylebar:

- Currency
- Percentage
- Thousand separator
- Decimal places—add and remove

How to Alter a Numeric Format

- 1 Select the numeric field to which you want to apply a format mask.
- 2 Choose the corresponding format mask button on the stylebar.

The format mask is applied.

Note: Oracle Developer applies the format mask only to fields whose datatype is NUMBER.

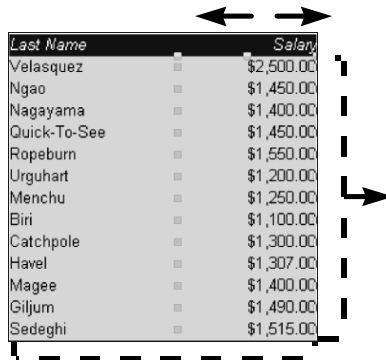
NLS Support for Format Masks

The stylebar always shows “\$” for the currency button, “,0” for thousands, and “.0” for decimal places, but the output currency symbols are language-specific, and can differ at run time.

If you use National Language Support, the symbols that appear in the report output depend on the setting of the NLS_LANG parameter.

Manipulating Objects

- Clear fields
- Move fields
- Resize fields



A screenshot of a report table with two columns: 'Last Name' and 'Salary'. The table contains 13 rows of data. Above the table, there are two horizontal arrows pointing left and right. To the right of the table, there is a vertical dashed line with an arrow pointing right. Below the table, there is a horizontal dashed line with an arrow pointing down. A small icon of a report with a plus sign is located to the right of the bottom dashed line.

Last Name	Salary
Velasquez	\$2,500.00
Ngao	\$1,450.00
Nagayama	\$1,400.00
Quick-To-See	\$1,450.00
Ropeburn	\$1,550.00
Urguhart	\$1,200.00
Menchu	\$1,250.00
Biri	\$1,100.00
Catchpole	\$1,300.00
Havel	\$1,307.00
Magee	\$1,400.00
Giljum	\$1,490.00
Sedeghi	\$1,515.00

- Flex mode adjusts layout during changes.



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Manipulating Objects

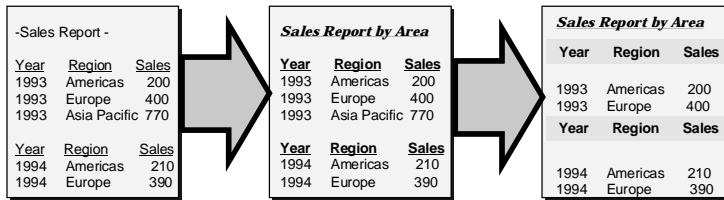
You can alter the position of objects, such as fields and labels, by selecting the object you want to move and dragging it to a new position.

Alter the size of an object by selecting one of the square handles and dragging to the required size, either smaller or larger.

Flex Mode When you move or resize an object, the rest of the report adjusts as necessary. This is controlled by the Flex Mode option, which is a button in the stylebar. The default is that Flex Mode is enabled, so that all report objects *flex*, or adjust, to make room for your modifications.

Modifying Visual Attributes

- **Modify text colors and fonts**
- **Edit text**
- **Set fill, patterns, and colors**
- **Use the Parent Frame tool to select an enclosing object**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

How to Change Field and Boilerplate Font

- 1 Select one or more objects.
- 2 Use the font pop-up lists in the stylebar to change font face and size.

How to Edit Boilerplate Text

- 1 Select the text you want to edit.
The text object is highlighted.
- 2 Click in the text to position the cursor in the text editor.
- 3 Edit the text as required.
- 4 Select any area outside the text object to accept the change.

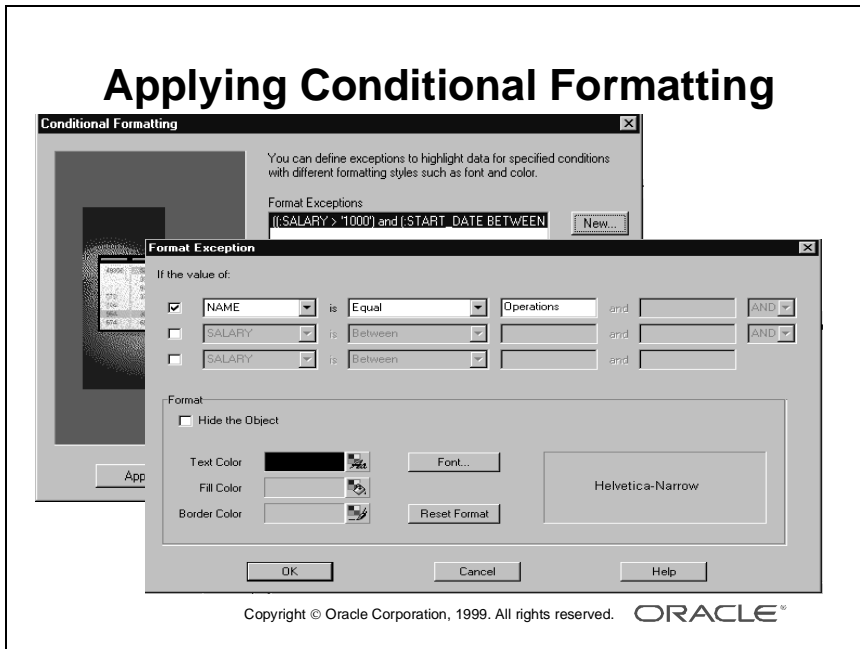
How to Modify Color Fills and Lines

- 1 Select the object that you want to modify.
- 2 Use the color palettes in the toolbar to change the fill, line, or text color.

The stylebar includes a Parent Frame tool that allows you to select the outer object surrounding the currently selected object.

To color the area surrounding all the labels, select one label, click Parent Frame, then select the color and pattern you require from the first color palette.

Applying Conditional Formatting



Conditional Formatting

You can highlight or suppress objects using conditional formats.

How to Create Format Exceptions

- 1** Select an object.
- 2** Display the pop-up menu and choose Conditional Formatting, or choose Format—>Conditional Formatting from the menu.
- 3** Create a new format exception.
- 4** Specify the conditions as required.
- 5** Set the format attributes as required.
- 6** Create further format exceptions as required.
- 7** Choose OK.

Formatting is applied in the order that the format exceptions are listed; a format exception overrides a previous format exception if both evaluate to TRUE.

You can select a format exception in the Format Exceptions list and move it up or down. You can also drag and drop items in the list.

Inserting Page Numbers, Dates, and Times

- Inserted easily
- Customizable extensions



<i>Last Name</i>	<i>Salary</i>
Quick-To-See	\$1,450.00
Nagayama	\$1,400.00
Magee	\$1,400.00
Giljum	\$1,490.00
Sedeghi	\$1,515.00
Nguyen	\$1,525.00
Patel	\$795.00
Dumas	\$1,450.00

Before

Tuesday, March 25 1997

<i>Last Name</i>	<i>Salary</i>
Quick-To-See	\$1,450.00
Nagayama	\$1,400.00
Magee	\$1,400.00
Giljum	\$1,490.00
Sedeghi	\$1,515.00
Nguyen	\$1,525.00
Patel	\$795.00
Dumas	\$1,450.00

Page 1 of 2

After

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Inserting Page Numbers, Current Date, and Time

Previewer Button	Description
Insert Page Number	Use this button to add a page number to the margin of a report.
Insert Date/Time	Use this button to add the current date and time to the margin of a report. You can define your own date and time format masks.

Date and Time Format Masks The Insert Date and Time dialog box displays the current date in many different formats. You simply choose the style that you want. The underlying format mask is composed of tokens representing each element.

The list of formats depends on the entries in your preferences file. To modify the preferences list, choose Tools—>Preferences—>Edit Masks.

Customizing the Date and Time Format If the mask that you want does not exist, and you do not want to add it to your preferences, choose Custom. This dialog box provides a read-only list of tokens available for you to build your own style, if you understand the Oracle format tokens.

Note: The Help system fully describes the format mask syntax. Choose Help—>Find—>Date Syntax, and choose Date and Time Format Mask Syntax from the list of topics.

Summary

- **Enhance report output using live data**
 - **Move, resize, delete objects**
 - **Edit text**
 - **Alter fonts, colors, and patterns**
 - **Apply format masks**
 - **Add page numbering and current date**
- **Save report to preserve changes**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Features of the Live Previewer

The Live Previewer is a WYSIWYG editor that allows you to enhance your report output using the live data. You can manipulate objects and edit text content. You can also alter the font, colors, and patterns of individual objects.

The format mask buttons provide quick and easy currency formatting. The page and date buttons allow you to add page numbering and the current date, selecting predefined formats or customizing your own.

Saving Changes

Remember to save your report definition often during editing, in order to preserve the changes you make in the Live Previewer.

When you save the report, all changes become part of your report definition and always appear at run time in the future.

Practice 5 Overview

**Modifying reports by applying changes in
the Live Previewer**

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**[®]

Practice Session Overview: Lesson 5

This practice session involves modifying reports by applying changes in the Live Previewer

Introduction

In this practice session you will reuse some of the reports that you created in the last session.

Practice Session: Lesson 5

- 1 Open report `p4q2.rdf`. Make the following changes in the Live Previewer:
 - a Make the columns SALARY and COMM right justified. Do not forget to justify the column header accordingly.
 - b Add commas and a currency symbol to SALARY. Add two decimal places.
 - c Add one decimal place and a percent sign to COMM.
 - d Edit the heading of the COMM column. Change the text to Comm Pct.
 - e Make the total at the end of the report right justified, and add commas, currency symbol, and two decimal places as in the column SALARY.
 - f Change the label of the total to bold font.
 - g Make whatever other changes you like.
 - h Save the report to a file named `p5q1.rdf` and close it.
- 2 Open report `p4q3.rdf`. Make the following changes in the Live Previewer:
 - a Remove the border around the total for each department.
 - b Add commas and two decimal places to the SALARY field and the total and make them right justified. Do not forget to justify the column header for the SALARY field accordingly.
 - c Make the same changes to the total at the end of the report. Move the grand total so that it aligns with the SALARY field. (You must do this manually; use the ruler guides to help you.)
 - d Change the color of the heading to gray.
 - e Make whatever other changes you like.
 - f Save the report to a file named `p5q2.rdf` and close it.
- 3 Open report `p5q3_a.rdf`. Make the following changes in the Live Previewer:
 - a Add a date at the top center of the page. Give it any format you want.
Hint: Choose the font, style, and text color you want before inserting the date.
 - b Make whatever other changes you like.
 - c Save the report to a file named `p5q3.rdf` and close it.

- 4** Open report `p4q6.rdf`. Make the following changes in the Live Previewer:
 - a** Right justify the `PRODUCT ID` field.
 - b** Remove the border around the cell values and the parent frame.
 - c** Add commas and two decimal places to the cells and summaries and make them right justified.
 - d** Add a gray line color to the parent frames of `NAME` and `PRODUCT ID`.
 - e** Save the report to a file named `p5q4.rdf` and close it.

If You Have Time

- 5** Open the file `p5q2.rdf` and conditionally highlight the `SALARY` field.
 - a** If the salary is less than 1000, then print the field in italics.
 - b** If the salary is more than 2000, then print the field in italics and make the text color red.
 - c** Save the report to a file named `p5q5.rdf` and close it.

6

Managing Report Templates

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe the template regions**
- **Describe the difference between default and override template sections**
- **Modify a predefined report template**
- **Register a customized template in the predefined template list**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

In this lesson you learn how to open and modify predefined Oracle Developer report templates and how to register a customized template in the predefined template list.

Objectives

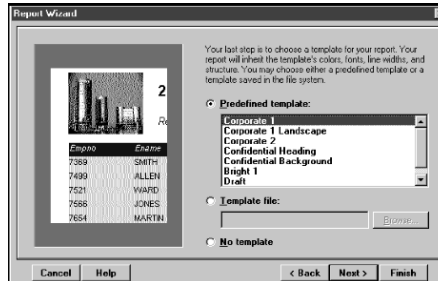
At the end of this lesson, you should be able to do the following:

- Describe the template regions
- Describe the difference between default and override template sections
- Modify a predefined report template
- Register a customized template in the predefined template list

Using Report Templates

Three template options:

- Predefined
- User-defined
- No template



Installation templates in

`<oracle_home>\report60\admin\template\us`

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Using the Report Templates

Introduction

The template page of the wizard has three options for choosing a template:

Option	Description
Predefined	Shows a static descriptive list of available templates
Template File	A user-defined template file that you specify by entering a filename or browsing the file system
No Template	Uses the report default settings instead of a template file

Creating a User-Defined Template

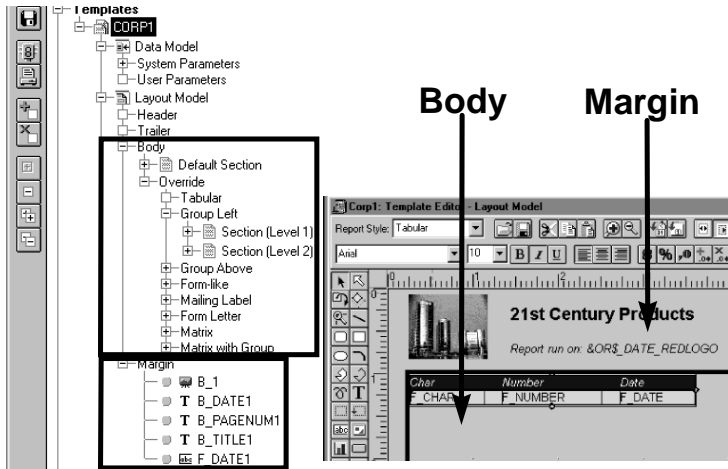
You can create your own templates from scratch, but this is not necessary, because many prebuilt templates exist in the installation.

Create your own templates by modifying one or more installed templates with your own standards, logos, and so on, as described on the following pages.

Several templates are included in the installation. They are in `<oracle_home>\report60\admin\template\us`. The file extension for templates is `.tdf`.

One template definition can contain template information for all styles of reports.

Modifying a Template



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying a Template

Introduction

A report template is not the same as a report definition. A template is divided into regions, which the Report wizard references while creating the report definition.

The Template Regions

The Template editor allows you to modify existing objects in the template's margin and body regions. You can also customize or add other reports objects: parameters, report-level triggers, program units, and attached PL/SQL libraries.

Region	Description
Body	Contains default section to apply changes to all report styles and override sections to apply individual changes to different report styles
Margin	Defines objects that appear in the margin of all pages

Making Modifications to Template Objects

Modify template objects in the Template editor or in the Object Navigator.

The Template editor is similar to the Layout Model view of the Report editor. You can create, delete, and modify objects (such as page numbers, text, and graphics) in the margin area. In the body area, you can modify the properties of body objects.

The Report Style drop-down list allows you to view the layout for a selected report style.

Default and Override Sections

Make general modifications to the default section. These modifications affect all report styles. However, you can override attributes for one or more styles to provide a flexible inheritance model.

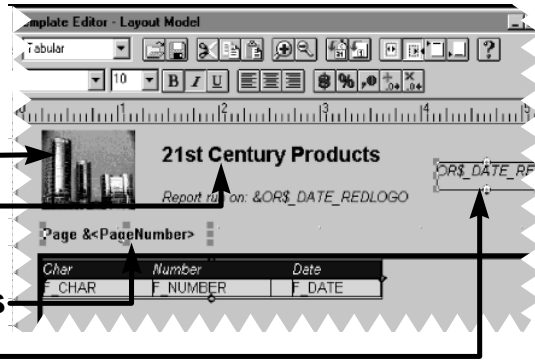
Each report inherits attributes from the template's default section unless there is an override value. The override section overrides specific attributes for each style.

There is a reinherit button in the property palette to return to the default value.

Customizing Template Margin

Include objects such as:

- Images
- Text
- Page Numbers
- Date



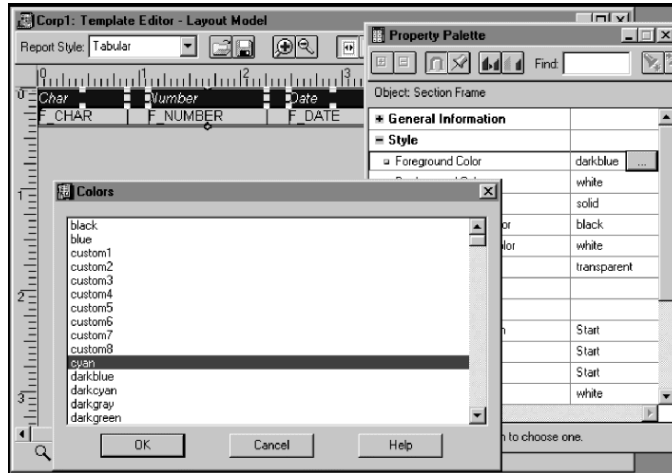
Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Customizing Template Margin

You can copy, create, or import various objects into the margin of the template.

- Images: File—>Import—>Image
- Text: Text tool in vertical toolbar
- Page Numbers: Toolbar button
- Date: Toolbar button

Customizing Template Body



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Customizing Template Body

In the template body you can modify fonts, colors, and fills of the field, label, summary, and frame objects. You make changes directly in the Template editor or by using the property palette and pop-up lists corresponding to the object selected.

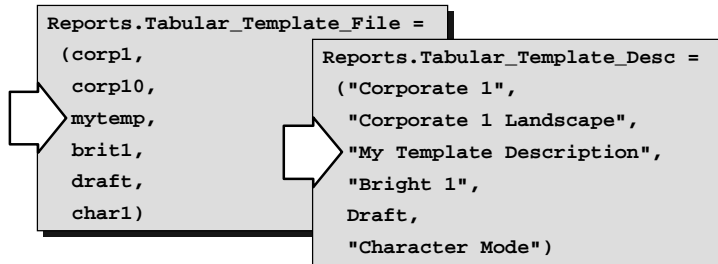
Reapplying a Template

If you reenter the Report wizard and reapply the template in the Template tab, Report Builder reapplies the template background and margin objects to your report.

The report retains any live changes you made to the body, such as modifying the font or color of text or fields.

Predefining Your Own Templates

- **Modify and save template file (mytemp.tdf)**
- **Move file to template directory:**
`<oracle_home>\report60\admin\template\us>`
- **Register name, description in cagprefs.ora**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Predefining Your Own Templates

Introduction

When you create your own templates, you might want to include some in the predefined template list that appears when creating a report using the wizard.

You can then select the template from the descriptive list for all similar reports that you create, instead of searching through your file system to find the template file that you want.

This section describes how to register the template file as a predefined template by inserting an entry in the preferences file and storing the template and (optional) image in the correct template directory.

Editing the Preferences File

The Report Builder builds the list of predefined templates and their images, by reading template names from the Oracle Developer preferences file and mapping to similarly named `.bmp` files (with a one-letter suffix to identify each report style).

How to Register a Customized Template in the Wizard's Predefined List

- 1 Open an existing template, modify the template objects as required, and save the template as a `.tdf` file.
- 2 Move the template and the image files to the template directory:
`<oracle_home>\report30\admin\template\us>`.
- 3 Add the template file name (excluding the suffix) to the entries in the Oracle Developer preferences file, `cagprefs.ora`. Add the file name to each style of report for which your template is applicable; for example:
`Reports.Tabular_Template_File, Reports.Matrix_Template_File.`
- 4 Add a short template description to the relevant `Reports.xxx_Template_Desc` entry in `cagprefs.ora`, making sure that the description is listed in the same position as the relevant file. These are the descriptions that you see in the predefined list when you apply a template in the Report wizard.

Adding a Template Image

- Image area is 200 x 150 pixels
- Move `.bmp` files to template directory
- Image file names = template name + one letter
- Example: template = `corp1`
 - Tabular image = `corp1t.bmp`
 - Matrix image = `corp1x.bmp`

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Adding a Template Image

Each predefined template has a small image associated with it that is displayed on the left side of the template page of the wizard. The image is optional but is a useful graphical example of how the report looks if you apply that template.

To display your own image, use the template to create a simple report and create an image of the report output.

How to Create a Template Image

- 1 Create a simple report using your new template. For example, use the tables EMP and DEPT.
- 2 Create a bitmap image (screenshot) of the top left corner, including enough data to make the style clear to the user.
Note: The bitmaps supplied are 200 x 150 pixels.
- 3 Name the file *.bmp and store it in the same directory as the template:
<oracle_home>\report30\admin\template\us>.

Template Image Naming Convention

Your image filenames must have the correct name for each style that you create. The image filename is the same as your template file plus one letter, depending on the report style.

Template Name	Image Name	Report Style
corp1	corp1a.bmp	Group Above
	corp1f.bmp	Form
	corp1g.bmp	Matrix with Group
	corp1l.bmp	Group Left
	corp1m.bmp	Mailing Label
	corp1r.bmp	Form Letter
	corp1t.bmp	Tabular
	corp1x.bmp	Matrix

Summary

- **Modify existing templates to create your own user-defined template**
- **Modify the Default section for all styles or Override for individual style changes**
- **Add your templates to the predefined list**
- **Display an image for each predefined template**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Modifying a Template

In this lesson you learned to open an existing template definition in the Template editor and make modifications to create your own customized templates.

You can use any customized template when creating a report in the Report wizard.

The template contains default and override sections.

- Default section: Modify objects to affect all styles of report created with this template.
- Override sections: Modify objects to make individual changes to one or more styles, Tabular, Group Above, and so on.

Registering a Template

You learned how to register your templates so that they appear in the predefined list in the Report wizard. Optionally, you can include a small bitmap image for each style of report, which is displayed when you select a predefined template.

Practice 6 Overview

- **Customizing an existing template**
- **Applying the modified template to an existing report**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 6

This practice session contains:

- Customizing an existing template
- Applying the modified template to an existing report

Introduction

In this practice session you open an existing template, which is one of the standard templates shipped with the Oracle Developer Release 6.0 installation.

You modify objects in the margin and body regions and save the template to a new filename. Then you apply the modified template to an existing report and note the differences.

Practice Session: Lesson 6

- 1 Customize an existing template, adding a company logo and report title to the report margin, and modify the report body.
 - a Open `corp1.tdf`. This file should exist in your home directory; it is a copy of the Corporate 1 predefined template.
 - b Display the margin region. (This usually appears by default when you open a template.)
 - c Delete the Oracle buildings logo. Import `summitlo.tif` and place in the top left corner of the margin; this file should exist in your home directory.
 - d Modify the report title line: Change 21st Century Products to Summit Report.
 - e Display the body region to view the changes you will make in the next steps. Make the changes general, not specific to a particular layout.
 - f Change the foreground color (fill color) of the section frame to white.
 - g Change the justification of number fields to End. Do not forget to change the justification of labels for number fields to End.
 - h Change the justification of number summary fields to End.
 - i Save the template as `summit.tdf`.
- 2 Apply the modified template to an existing report.

Hint: You must search for the template file; it does not appear in the list of predefined templates.

- a Open `p4q3.rdf`.
- b Apply `summit.tdf` to the report.
- c The report shows some changes but not all. Which template changes have been applied, and which have not?
- d To apply all changes to your report you will have to create the layout again. From the menu choose Edit—>Select All to select all layout objects. Now delete them. Invoke the Report wizard and reapply your template.
- e Save the report as `p6q2.rdf`.

-
- 3 Continue to modify the template and test it using a new report.
- a Change the text color of the master group fields to red for the Group Left style only.

Hint: In the Template editor, use the Report Style list to switch to a Group Left style. This way you can select the correct group. You may also want to use the Zoom Out tool, so that you can see the entire template layout structure, and recognize the difference between the master group fields and the detail group fields.

- b Save your template as `summit2.tdf` and close it.
- c Create a new report using the Report wizard.
Use the layout style Group Left.
Import the query `p6q3.sql` from your working directory.
Make NAME the group field.
Display all other columns.
Do not create summaries.
Change the width of SALARY to 6 and COMM to 4.
Apply the Template file `summit2.tdf`.
Save the report as `p6q3.rdf` and close it.

Report Storage Methods

Objectives

At the end of this lesson, you should be able to do the following:

- **Distinguish between different storage methods of report definitions**
- **Compare report file types and their portability**
- **Convert reports to different storage types**
- **Upgrade report and printer definition files from Developer/2000 Release 1**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

When you save a report definition, you have the choice of storing it in the file system or in the database. In this course we use the file system.

This lesson identifies the various methods of storing report definitions. It also covers the different file types associated with report building, and portability issues that apply in cross-platform deployment.

Objectives

At the end of this lesson, you should be able to do the following:

- Distinguish the different storage methods of report definitions
- Compare report file types and their portability
- Convert reports to different storage types
- Upgrade report and printer definition files from Release 1

Storage Methods

- **Options:**
 - File system
 - Database
- **Considerations:**
 - Application management
 - Storage restrictions

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report File Types

Type	Description
.rdf	Full report definition (includes source code and comments) modifiable through Builder binary, executable
.rep	No source code or comments not modifiable binary, executable
.rex	Full report definition not modifiable ASCII text, not executable

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Storage Methods and Requirements

Introduction

You can store any Report Builder module, report, external query, or external PL/SQL library either in an operating system file or in the Report Builder database tables.

The most common method, and often the most efficient depending on your network speed, is in a system file. However, your choice depends on your own system of application management (how you handle file security and version control) and any restrictions, in terms of database size and file storage allocations.

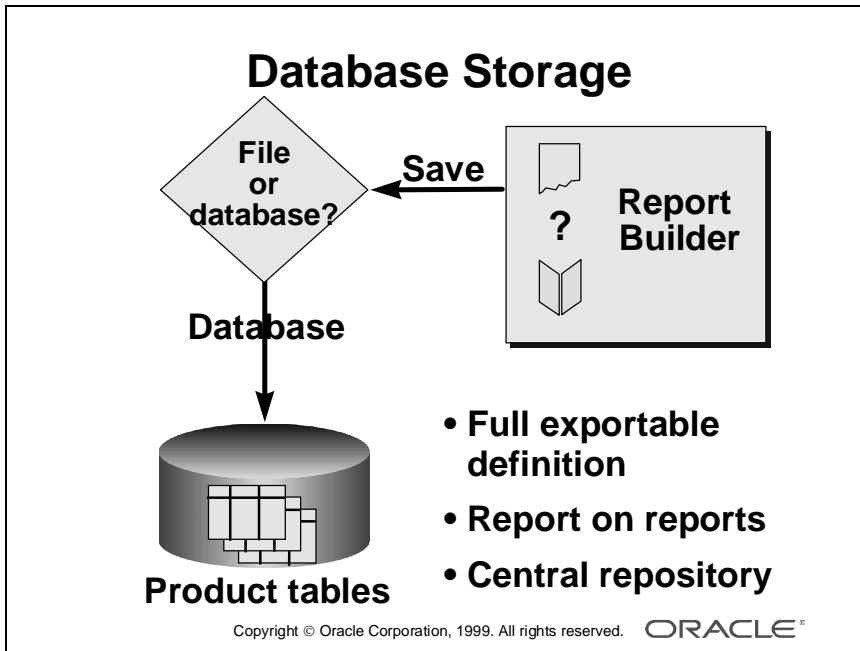
Note: You can store template modules in an operating system file only, not in the database tables.

Operating System Report File Types

Any report you save to a file has the file extensions and properties listed in the table below.

File Type	Contents	Creation	Format	Features
.rdf	Single report definition, including source code and comments	File—>Save File—>Save as File—>Administration —>Convert	Binary executable, modifiable in Report Builder	Full report definition, and executable in one file
.rep	Single runnable report, no comments or source code	File—>Administration —>Compile Report File—>Administration —>Convert	Binary executable, not modifiable	Reduced size of executable, prevents user modification
.rex	Single report definition, including source code and comments	File—>Administration —>Convert	Text (ASCII or EBCDIC), not executable, not directly modifiable (convert to .rdf and modify in Report Builder)	Source control and can be used for debugging

For a full list of supported files, in addition to report files, see the online Help: Contents—>Report Builder Reference—>Storage—>Supported Files



Database Table Storage

When you save a module to the database, its definition is stored in the Report Builder tables. These tables hold a combination of report-specific and integrated Oracle product information. As with SQL*Plus and accessing the data dictionary, you see the data through views.

Why Save to the Database?

You may want to save reports to the database for these reasons:

- Export from one database to another
- Document your reports
- Modify the Report on Reports .rdf files, which run against the Report Builder product tables
- Use as a central repository during the build process

Installing Database Tables

In order to save definitions to the database, the Report Builder product tables must exist in the database, userid SYSTEM. Usually a system administrator installs these tables for the development team.

These product-specific tables form a repository to store all the information about your report definitions.

Reference

For further information, see the online Help:

- Manuals—>Oracle Developer 6.0: Getting Started—>Building and Maintaining the Database Tables.
Help Topics—>Report Builder Reference—>Storage
- Help Topics—>Report Builder Reference—>Administration

Documenting Reports

- **Run Report Builder utility**
File—>Administration—>Report on Reports
- **Document multiple reports using wildcards:**
% and _
- **Modify installation reports**
 - <oracle_home>\report60\srwdoc1b.rdf>
 - <oracle_home>\report60\srwdocpb.rdf>
- **Run batch mode: RWRUN60...BATCH=YES**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Documenting Your Reports

Introduction

The most common reason for storing your reports in the database is so that you can document your report definitions for maintenance or debugging purposes.

- Use the Oracle Developer facility in Report Builder to produce documentation describing your report definition.
- Write your own reports to query the Report Builder tables.

How to Create Documentation in Report Builder

- 1 Save one or more reports to the database.
- 2 Select File—>Administration—>Report on Reports.
- 3 Choose Portrait or Landscape orientation for your report.
- 4 Fill in the run-time parameter form and choose OK.

Note: If the Report Builder tables do not exist, Oracle Reports displays an error message after Step 2.

Documenting More Than One Report




You can document more than one report by using wildcards, % and _, in the report name. You can also choose to omit all comments and/or PL/SQL code, and specify the level of detail you require—for example, document the Data information or Layout information only.

Invoking the Documentation Reports

The Report on Reports menu item runs two reports, `srwdoclb.rdf` for landscape orientation and `srwdocpb.rdf` for portrait orientation. They exist in the Report60 product directory, usually `<oracle_home>\report60`.

These reports are Report Builder definitions. You can modify them by opening them in Report Builder.

You can choose to document a number of reports in batch, by invoking either of these reports with the `RWRUN60` command, including the run-time parameter `BATCH=YES`.

	File Portability
.rex	Portable as text
	Portable as binary
.rd	
f	
	Sometimes portable as binary (if no PL/SQL)
.rep	

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

File Portability

Introduction

You may need to move report definition files from one machine to another, either during development or when installing the production application.

Portability

Each file created has different characteristics that determine how portable it is across platforms. The table below lists the portable files, with any limitations they may have:

File Type	Portability
.rex (text)	100% portable
.rdf (binary + text)	Portable if transferred as binary (PL/SQL recompiles on open and run)
.rep (binary)	Portable as binary if no PL/SQL is present

Although .rep files are considered portable when they do not contain PL/SQL, it is rare to have a report that does not contain any PL/SQL code. Oracle recommends that you always consider using either .rdf or .rex files when porting, and recompile .rep files for each operating system.

File Searching Method

**Hardcoded path that prefixes file name
or**

- 1. Path that the operating system considers the current path**
- 2. Path specified in SOURCEDIR (RWMTS60 only)**
- 3. Path specified in REPORTS60_PATH**
- 4. Path specified in ORACLE_PATH**
- 5. Location when Report Builder or Report Runtime was invoked**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

File Searching Method

When you reference a file in Reports, Oracle Reports searches for the file as follows, depending on whether you include the pathname in the reference:

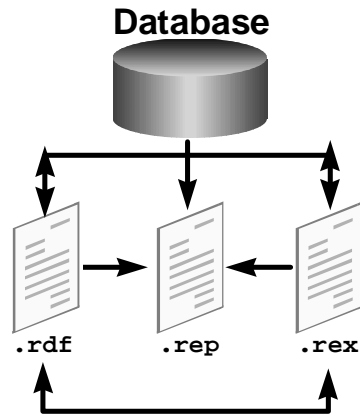
- The hardcoded path prefixing the filename (if specified)
- The following variable paths in sequence:
 - 1 The path that your operating system considers the current path
 - 2 The path specified in the environment variable `SOURCEDIR` (searched by Reports Server, `RWMTS60`, only)
 - 3 The path specified in the environment variable `REPORTS60_PATH`
 - 4 The path specified in the environment variable `ORACLE_PATH`
 - 5 Your location when you invoked Report Builder or Report Runtime

Note: Oracle Reports uses the variable paths to locate reports and external files such as PL/SQL libraries, external queries, and external boilerplate only if the path is *not* hardcoded.

For further information on the subject, see the online Help system.

Converting Report Definitions

Invoke RWCON60 from Report Builder menu, Windows menu, or command line



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Converting Report Definitions

Introduction

You can convert your report definitions from one storage type to another by using the RWCON60 executable.

Invoking the Conversion Utility

You can invoke this conversion in one of three ways:

- From Report Builder: File—>Administration—>Convert
- From the Windows menu: Start—>Programs—>Oracle Developer 6.0—>Report Compiler
- From the command line: RWCON60 + arguments

Conversion Options

The following report conversions can be performed using RWCON60:

- Convert a report stored in the database into a *.rep*, *.rex*, or *.rdf* file
- Convert a report stored in an *.rdf* file into a *.rep* or *.rex* file, or a database report
- Convert a report stored in a *.rex* file into a database report, *.rdf* file, or *.rep* file

Note: You can also convert a report definition to a template definition. However, Oracle does not recommend this as a method of creating your templates, because the information stored in the converted template is limited; only Margin objects are converted.

Migrating Reports from Previous Releases

- **Open and save reports in Release 6.0**
 - **Automatic PL/SQL upgrade utility**
 - **Log file contains syntax incompatibilities**
- **Use wizard for major modification**
- **Use layout model for individual changes**
- **Move printer definitions to**
`<oracle_home>\report60\printers>`

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Migrating Reports from Previous Releases

Introduction

You can migrate existing report definitions from Oracle Developer Release 1 (Reports 2.5) and Oracle Developer Release 2 (Reports 3.0) by opening and saving the report in the Oracle Developer Release 6.0 Report Builder.

Upgrading PL/SQL Code

Oracle Developer	PL/SQL
Release 1	PL/SQL 1
Release 2	PL/SQL 2.3
Release 6	PL/SQL 8

Because of syntax differences between the PL/SQL versions, an upgrade utility automatically verifies your PL/SQL syntax when you first open an older report definition.

The upgrade utility reports any syntax incompatibilities that it cannot correct, and creates a log file for you to view the messages and manually alter the code as necessary.

Modifying Migrated Reports

You can reenter the wizard to make changes to an upgraded report, but remember that the wizard recreates your layout, so you lose all layout changes, including layout triggers, that existed in the original report definition.

To make individual changes to upgraded reports, use the Layout Model editor, which also provides an additional default layout tool.

Migrating Printer Definition Files

The older Oracle Developer printer definition files (.prt), used for character-mode reports only, are totally compatible with Oracle Reports Release 6.0. Copy your customized printer definition files to the printer directory:

```
<oracle_home>\report60\printers.
```

Summary

- **Storage methods: file or database**
- **Report on reports documentation utility**
- **File portability**
- **Conversion between storage types**
- **Upgrading report and printer definitions from previous releases**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

In this lesson you learned the difference between saving a report definition to a file, including the three different report file types, and saving in the database.

One reason for saving in the database is to use the Report on Reports utility to document your report definition.

Most Oracle Report Builder files are portable, some as text, others as binary.

Use the Reports Convert utility to convert between the different report storage types.

Upgrading from Release 1 does not require a special utility. Open a Release 1 report in Report Builder. If the report definition contains PL/SQL, the code is automatically upgraded to PL/SQL 8.

Practice 7 Overview

- **Using the Help system to find information on storage types and executables**
- **Converting a report using the Convert utility**
- **Documenting a report using the Report on Reports utility (optional)**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 7

This practice session contains:

- Using the Help system to find comprehensive information on storage types and executables
- Converting the storage type of a report using the Convert utility
- Documenting a report using the Report on Reports utility

Since the Report Builder tables are optional in Oracle Reports, your classroom might not have the tables available for you to save your report. Therefore your instructor will advise you whether to attempt this question.

Practice Session: Lesson 7

- 1 Search the Help system for information on the RWCON60 executable.
What is this executable used for?

Can you compile more than one report with this utility?

What other tool could you use to compile several reports?

Hint: You may want to refer to Lesson 1.

When converting a report to a template, what is converted and what is not?

- 2 Use the Convert utility in Report Builder to convert a report from *.rdf* to *.rep*.
 - a Open `p6q3.rdf`, if it is not already open.
 - b Convert it to a `.rep` file.
 - c What other methods could you use to compile a report?

What is the difference?

- d Convert the same report to a `.rex` file.
- e Open a system editor to look at the `.rex` file.

If You Have Time

- 3 If the database product tables are installed, document a report definition that is saved in the database.
 - a Save report `p6q3` to the database.
 - b Run the Report on Reports utility to produce documentation on the report `p6q3`. Set the run-time parameters to output to a file in PDF format.
 - c Open the PDF file and scroll through the documentation.

**Enhancing Reports Using
the Data Model: Creating
Queries and Groups**

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe Data Model objects.**
- **Modify report hierarchy.**
- **Change the order of data in a group.**
- **Eliminate data from a report.**
- **Create supplemental rows of data, by adding queries.**
- **Link data from different queries.**
- **Reuse PL/SQL code to create a Ref cursor**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Overview

Introduction

You have learned to create simple, single-query reports using the Report Wizard. For many reports, one query is sufficient. However, sometimes reports require multiple queries and a more complex structure than the Report Wizard provides.

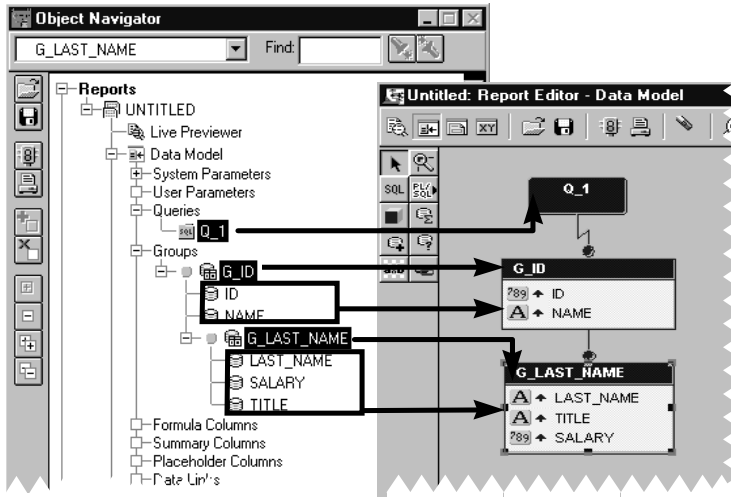
In this lesson you learn the Data Model structure and objects, and how to enhance reports by creating additional groups and queries to achieve more complex report structures.

Objectives

At the end of this lesson, you should be able to:

- Describe the Data Model objects and their relationship
- Modify the report hierarchy by creating additional groups
- Change the order of data in a group
- Eliminate data from the report using a group filter
- Create supplemental rows of data by adding queries
- Link data from different queries by creating a data link

The Data Model Objects



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Describing the Report Data Model

Introduction

The Data Model defines the report data and its hierarchy—that is, the report structure. The Data Model does not define any formatting attributes for the report output. This section introduces you to the objects in the data model and their appearance in the Data Model editor.

The Data Model Objects

The Data Model consists of the objects listed below.

Object	Description
Query	You can create a report with one or more queries. The Report Wizard creates a single-query report. Use the query tool or the Data Wizard to create additional queries.
Group	Each group is owned by a query. By default, Reports creates one group for each query.
Column	Columns are owned by the group to which they belong. By default, the group contains one column for each select expression in the query.
Link	Links join a parent group to a child group. You can create links to form a relationship between groups from different queries. Links are <i>never</i> created by default.
Parameter	Parameters are owned by the report. You can create parameters that allow users to enter value restrictions at run time. Reports also provides a number of system parameters by default.

What Does the Report Wizard Create?

The objects that the Report Wizard creates by default depend on the style of report that you choose.

- Tabular style: One query, one group
- Group Above, Group Left: One query, two or more groups

The number of group objects depends on the number of break groups you define in the Report Wizard.

The Report Wizard creates only one query.

Modifying Properties of a Query

- **Modify SQL Query Statement**
 - Add, rename, or delete columns.
 - Use column and table aliases.
 - Remove or modify schema name.

```
SELECT d.id, d.name, e.title,  
       e.first_name||' '|e.last_name employee_name,  
       e.salary*12 annual_salary  
FROM   s_emp e, s_dept d  WHERE e.dept_id = d.id
```

- **Syntax error checks occur when:**
 - Exiting SQL query statement
 - Compiling or executing a report

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Modifying Properties of a Query

Introduction

You can modify a query, including the SQL statement, by changing properties in the query property palette.

Editing the SQL Query Statement

To edit the query statement, open the property palette, choose the SQL Query Statement property, and click the left button.

The SQL Query Statement dialog box appears.

You can also use the Data Wizard to make changes to the query. The Data Wizard is covered later in this lesson.

Modifying Columns If you add to, rename, or delete a column or expression in the query statement, Reports automatically creates, revises, or deletes the corresponding column in the Data Model.

Using Column and Table Aliases Remember that you can use aliases against the database columns and tables in the query. Report Builder uses your column aliases to create the relevant report column names. Table aliases help your query access the database and return data as quickly as possible. Aliases therefore provide:

- Fast database access and return of data
- Shorter, more meaningful names for query expressions, such as `employee_name` and `annual_salary` in the example opposite

Dealing with Error Messages

Reports checks the syntax when you try to leave the SQL Query Statement dialog box and whenever you generate or execute the report.

If Reports detects an error, it displays the error in an Alert.

More Properties

Aid maintenance and testing

- **Rename queries in complex reports**
- **Add comments to describe queries**
 - **-- comment for single lines**
 - **/*comment*/ for multiple lines**
- **Set Maximum Rows to restrict data**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE[®]

Renaming Queries

Oracle recommends that you rename queries in a complex report, to make the Data Model easy to understand.

Renaming a query does not automatically rename the group below it.

Commenting Your Queries

Use the Comment property to enter meaningful text that describes the query.

You can also insert comment lines directly in the SQL Query Statement area, using the Oracle standard comment delimiters:

- `--comment` for a single line
- `/*comment*/` for multiple lines

Note: For ease of maintenance, do not enter comments in different places for the same query.

Restricting Rows

Set the Maximum Rows property to restrict the rows returned from a query. This is useful for testing your report against a large database.

Updating the Layout

- **Update layout to reflect changes in Data Model.**
 - **Choose Report Wizard.**
 - **Alter tab pages.**
- **Wizard destroys previous layout and creates new objects.**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Updating the Layout

When you alter the Data Model, you must update your layout to see the changes.

Choose Report Wizard from the toolbar, or Tools—>Report Wizard from the menu, and alter the tabs to create a new layout with additional groups and columns.

Note: If you edit a query statement to alter any columns that are part of an existing layout, and run the report without recreating the layout, the Source property becomes null, and the field in the previewer appears with a large cross.

Groups

Groups determine hierarchy and frequency

- **Wizard creates default groups**
 - **Default naming conventions**
 - **You can change query name**
 - **You can change group name**
- **Developer-created groups for**
 - **Control break reports**
 - **Complex matrix reports**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Changing the Group Structure

Introduction

Groups organize your data into sets of records and, in a hierarchy, determine the frequency at which records in that group appear in the output. Each group contains columns that Report Builder assigns to the group by default, or that you specifically assign.

Changing the Group Name

If you change a query name after the group has been created, the group name *does not* change automatically.

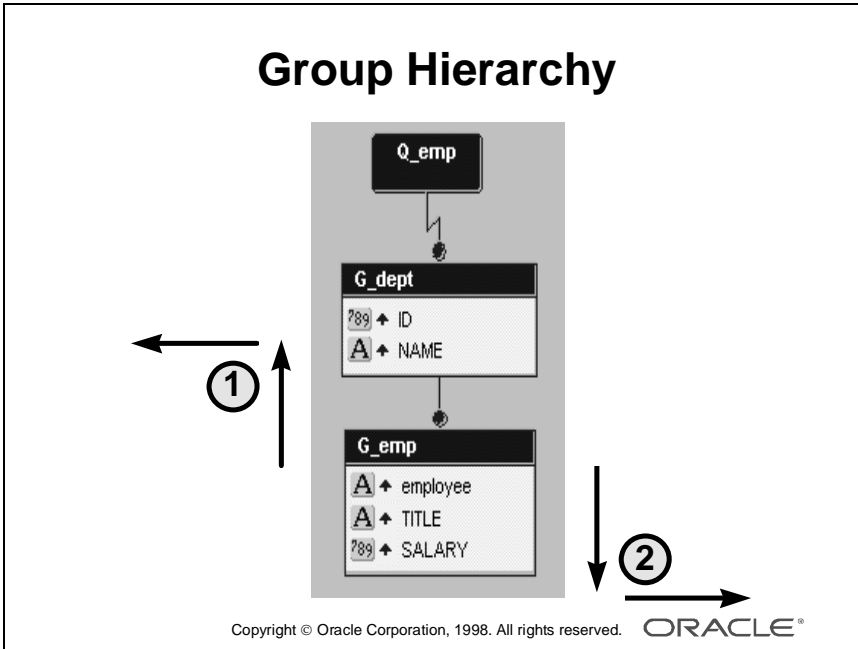
There are two ways you can change the group name:

- Open the Group property palette and change the Name property.
- In the Object Navigator, select the node representing the group and then select the group name again. This removes the highlight and changes the cursor to an I-beam. Edit the group name. The next time you activate the Data Model editor, the new group name is visible.

Developer-Created Groups

Sometimes you do not want to modify existing Data Model objects in the Report wizard; for example, during later maintenance. In this case, you might need to create your own groups in the Data Model to perform the following actions:

- Produce control break (nested) reports
- Produce complex matrix reports



1	Drag columns up or left, to create a parent group.	2	Drag columns down or right, to create a child group.
---	--	---	--

The Group Hierarchy

Introduction

The groups in Reports exist in a hierarchy. You can alter the group hierarchy by creating new groups.

As you have already seen, a single query produces one default group. You can produce a *control break report* structure by creating one or more additional groups above the default. The following example of a single query, called Q_EMP, is the basis for the hierarchical report.

```
SELECT  e.first_name, e.last_name, e.title, d.id, d.name
FROM    s_emp e, s_dept d
WHERE   e.dept_id = d.id
```

Creating a New Group

Select D.ID and drag it out of and above the default group to create a break group. Change the group name to make it more meaningful.

You can drag other columns, such as D.NAME, into the new group. Alternatively, move more than one column into a new group by shift-selecting the columns before dragging them all together.

Note: You can also drag columns *down* a hierarchy, from a higher group into a new or existing lower-level group. The order in which the groups exist in the Data Model has an effect on the output. Always make sure that your parent (master) group is first, child (detail) group is second.

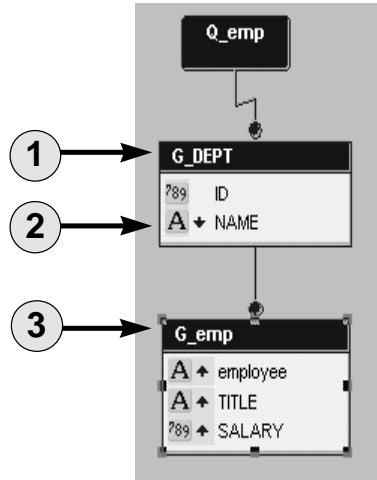
One-Level Limit

When you drag a column to create a new group, the new group is only one level above or below the column's original group in the hierarchy.

To create a group more than one level from the original group, follow these steps:

- 1 Move the column to the level immediately above or below the required position of the new group.
- 2 Drag the column out to create the new group at the next level.

Ordering Data in a Group



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

1	Break group controls primary sort; uses Set Break Order and Break Order properties
2	Order by NAME column: Set Break Order = Yes; Break Order = Descending
3	Child group controls final sort; uses ORDER BY clause in the query

Ordering Data in a Group Hierarchy

Introduction

Break groups that you create in a hierarchical report display data in a default order. You can modify this default.

Break Order

Break order controls the break structure and the order in which to display the column's values. It is denoted by a solid arrowhead to the left of the column. For column values in user-created groups, you must use the Set Break Order and Break Order properties to specify how to order the break column's values. The order of precedence depends on the order of columns, from the top down, in the group.

Reversing the Break Order

To reverse the break order, double-click the column in the group to open up the Column property sheet. Change the break order from Ascending to Descending using the pop-up list. In the Data Model, the arrowhead is displayed in reverse.

Note: All break groups must have at least one column with Set Break Order = Yes.

Break Order in Break Groups

Remove unnecessary break columns in each break group to improve the performance of your report. A break group is any group other than the lowest group of each query. If several columns are all unique, such as the department ID and NAME columns, having break order set on both columns can sometimes cause unnecessary sorting.

- 1 Open the NAME column property palette.
- 2 Alter Set Break Order to No.
- 3 Repeat for each column that is not part of the sort.

Break Order in Lower Groups

The Break Order property has no effect on columns in the lowest group of each query. Modify the ORDER BY clause in the SQL query statement to control this group.

Removing the Break Order property from columns in this lowest group is optional and has no effect on the output.

Query Modifications

In a break report, data order is determined by:

- Break order columns in the break groups
- Columns that you specify in the **ORDER BY** clause

```
SELECT d.id, d.name, e.last_name...  
FROM   s_emp e, s_dept d  
WHERE  e.dept_id = d.id  
ORDER BY 2, e.salary
```

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Reports Query Modifications

The examples below show how the ORDER BY clause is modified.

For each database column with Set Break Order = Yes, Reports appends an ORDER BY clause to the query or modifies the existing ORDER BY clause.

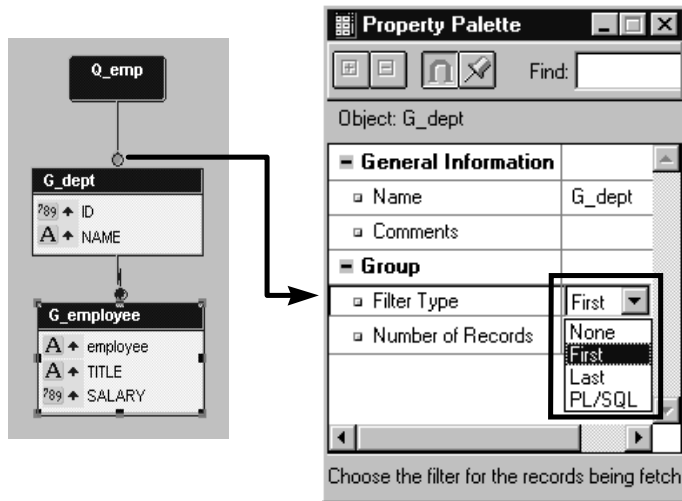
The break columns always take precedence over the lowest group columns. For example, suppose that your report contains the following query:

```
SELECT d.id, d.name, e.last_name, e.salary
FROM   s_emp e, s_dept d
WHERE  e.dept_id = d.id
ORDER BY e.salary
```

If you create a break group containing D.ID and D.NAME, but with D.NAME as the only break column, then at run time your query becomes:

```
SELECT d.id, d.name, e.last_name, e.salary
FROM   s_emp e, s_dept d
WHERE  e.dept_id = d.id
ORDER BY 2, e.salary
```

Filtering Data in a Group



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Using a Packaged Filter

Reports provides two packaged filters:

- **First**
Retrieves the first <n> records for the group
- **Last**
Retrieves the last <n> records for the group

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Filtering Data in a Group

Introduction

A *group filter* allows you to control the processing of records in each group. This section briefly describes the two types of filter available and how to use a packaged filter to restrict the number of records returned.

Two Types of Group Filter

- A packaged filter allows you to restrict processing to a predetermined number of records.
- A developer-created filter allows you to control processing dependent on conditional PL/SQL code.

Note: Developer-created filters are covered in more detail later in the course.

Identifying an Active Filter In the Data Model editor, the circle above the group object changes to green when a filter is active.

Using a Packaged Filter

A *packaged filter* allows you to specify the number of records that you wish to retrieve in a group. You can use this to restrict the data while developing your reports.

The Two Packaged Filters

- The *First filter* retrieves the first <n> records for the group.
- The *Last filter* retrieves the last <n> records for the group.

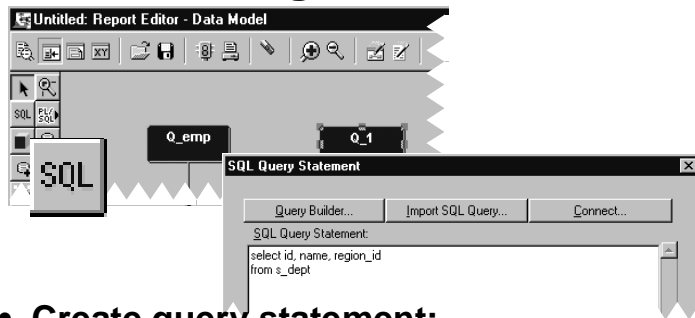
How to Apply a Packaged Filter

- 1 Double-click the group title bar, or its icon in the Object Navigator, to display the property palette.
- 2 Change Filter Type to First or Last.
The Number of Records property appears.
- 3 Enter the number of records you require.

How Is Data Fetched in a Packaged Filter?

- First filter: Reports retrieves a multiple of the array size, sufficient to satisfy the filter, based on a parameter value at run time.
- Last filter: Reports must retrieve all records to establish which are last.

Adding New Data



- **Create query statement:**
 - Enter valid syntax
 - Choose Query Builder
 - Choose Import SQL Query

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Adding New Data

Introduction

For many standard reports, one query is sufficient. However, reports sometimes require multiple queries. This section describes how to add data to your report by creating additional queries.

How to Create a New Query

- 1 In the Data Model editor, select Data Wizard from the toolbar.
- 2 Name the query as required.
- 3 Create the query statement in one of three ways:
 - Enter the syntax of a valid select statement.
 - Choose Query Builder and build the query.
 - Choose Import SQL Query and select a `.sql` file.
- 4 Create additional groups as required.
- 5 Create totals as required.

Creating a New Query Manually In the Data Model editor, select the Query tool and click in the Data Model painter area to create a new query.

A Query object appears and the SQL Query Statement dialog box opens.

Note: This behavior is controlled by a tools preference.

Create the query statement as in the Data Wizard.

Default Groups

Each query that you create in the Data Model creates at least one group. Report Builder creates a unique group name by appending the first column name in the group, `G_id`, `G_dept_id`, and so on.

Using an External Query

- **Import an external query from a file**
 - Import SQL Query button
- **Create an external query file**
 - Notepad
 - SQL*Plus
 - Report Builder
- **Link to an external query source**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Using an External Query

The Import SQL Query option allows you to import the contents of a text file that contains a valid SELECT statement. This file is referred to as an external query. The external query file contains one SELECT statement only.

Creating an External Query

You can create an external query in several ways:

- Open a text editor, such as Notepad.
- Invoke SQL*Plus. Enter, run, and test a query, then save.
- In Oracle Reports, choose File—>New—>External Query. You can access Query Builder from the Tools menu or the toolbar.

When creating an external query in Report Builder, you can save the query to the database instead of a file.

Linking to an External Query Source File

Instead of importing the contents of a file into your report definition, you can create a link to a separate source. The external query source can exist in a file or in the database (report product tables).

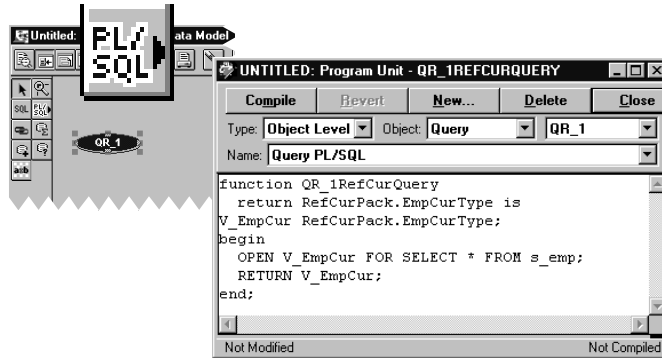
Select the External Query Source File property for the query and enter the name of a file. If the external query exists in the database, prefix DB: to the query name.

An external query source file can be a maximum of 32K.

Report Builder validates the external query source when you enter the name in the property palette and also at run time. Therefore, the following rules apply:

- The external query must always exist at run time.
- You must ensure that, if you update the external query, it does not invalidate any dependent report definitions.

Ref Cursor Queries



Using Ref Cursor Queries

Introduction

A ref cursor query uses PL/SQL to fetch data. Each ref cursor query is associated with a PL/SQL function that returns a ref cursor. The function must ensure that the ref cursor is opened and associated with a SELECT statement that has a SELECT list that matches the type of the ref cursor.

How to Create a Ref Cursor Query

- 1 Create a package that defines a ref cursor type.

```
PACKAGE RefCurPack IS
  TYPE EmpCurType IS REF CURSOR RETURN s_emp%ROWTYPE;
END;
```

Note: The package can be either a local program unit, part of a PL/SQL library, or a stored program unit.

- 2 In the Data Model editor, select the Ref Cursor Query tool and click in the Data Model painter area.
- 3 In the PL/SQL Editor, enter the PL/SQL function that opens a cursor and returns a cursor variable of the ref cursor type you defined in the package.

```
FUNCTION QR_lRefCurDS return RefCurPack.EmpCurType is
  V_EmpCur RefCurPack.EmpCurType;
BEGIN
  OPEN EmpCur FOR SELECT * FROM s_emp;
  RETURN V_EmpCur;
END;
```

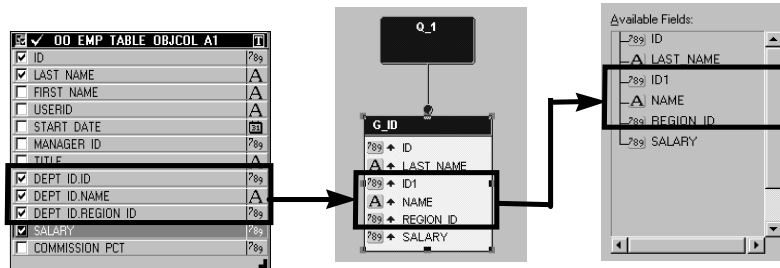
- 4 Refine the query properties as desired.
- 5 Define break groups as required, using the Data Wizard.

Reference

For further information on ref cursors see:

- Online Help: Help Topics—>PL/SQL Reference—>PL/SQL Language
- Elements PL/SQL User's Guide and Reference

Oracle8 Objects: Object Columns



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Oracle8 Objects in Report Builder

Introduction

Report Builder now supports basing a report on object tables, methods, tables or views with object columns, and tables with REF columns. Object tables are treated just like relational tables. The Database Objects node contains the schemas and their objects you have access to.

Object Columns in Report Builder

An object column is a column based on a user-defined object type definition. It has a composite rather than a scalar datatype. You can think of an object column as a record.

Query Builder

If a table contains an object column, Query Builder shows the table as a flat structure. You can select each of the attribute columns individually.

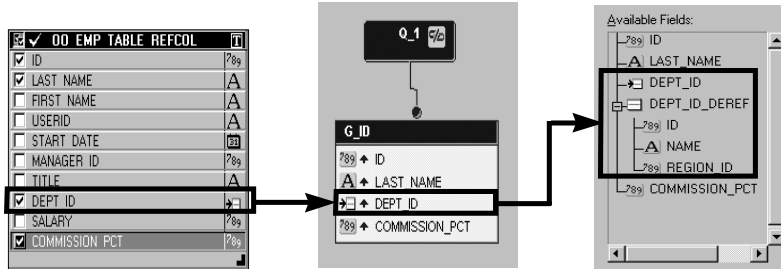
Data Model

If a table contains an object column, the Data Model shows the table as a flat structure. The attribute columns display as any other individual column.

The Wizards

The Wizards allow you to select individual components, the attribute columns are treated as any scalar column in the table.

Oracle8 Objects: REF Columns



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

REF Columns in Report Builder

A REF column does not hold data directly. It contains an object identifier (OID) that points to another table (object table) that holds the actual data. Although the attributes of the referenced object are not stored in the same table as the pointer, you need not select the referenced table. Report Builder treats the referenced fields as if they were nested objects.

Query Builder

Because the table does not contain a nested object, but only a pointer to another table, Query Builder shows only the REF column.

Data Model

The Data Model displays the REF column as any other column, except that it displays an icon for the datatype.

The Wizards

If your report definition includes a column of the type REF, several entries are added to the Available Fields list in the wizards.

- The REF column itself. You do not need to select this unless you want to display or use the REF value on the client side.
- A DEREf placeholder. The referenced object must be dereferenced to return the object identified by the reference. If you select this placeholder, all of the referenced fields will be selected.
- The referenced fields. You can select the fields individually, just like any other field.

Reference

For further information on Oracle8 objects, see:

- Online Help: Help Topics—>New Features—>Oracle8 Support—>Oracle8 datatypes supported
- Appendix F

Viewing Output from Multiple Query Reports

The screenshot displays the Oracle Report Editor - Live Previewer interface. On the left, a data model is shown with two queries: Q_emp and Q_job. Q_emp is linked to a group G_dept, which contains fields ID and NAME. Q_job is linked to a group G_job, which contains fields DEPT_ID, TITLE1, and count_id. Below these, a group G_LAST_NAME is shown with fields LAST_NAME, TITLE, and SALARY. The main area shows the report output, which is a table with columns Title and Salary. The data is grouped into three sections, each titled 'NameOperations'. The first section lists Warehouse Manager (1100), Stock Clerk (950), and Stock Clerk (750). The second section lists Warehouse Manager (1300) and Stock Clerk (800). The third section lists Warehouse Manager (1300), VP (1), Finance (1), Sales (1), Represent (1), ative (1), VP, Sales (1), Sales (1), and Represent (1). A small table at the bottom left shows the following data:

Dept Id	Title1	Count Id
10	VP,	1
31	Finance	1
31	Sales	1
31	Represent	1
31	ative	1
31	VP, Sales	1
32	Sales	1
32	Represent	1
32	ative	1

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Viewing Output from Multiple Query Reports

When you reenter the Report Wizard and update the layout based on multiple queries in the Data Model, the order in which unrelated groups of records appear in the output depends on the position of queries and groups in the Data Model.

It is the position of the group that determines the processing, not the query.

Example

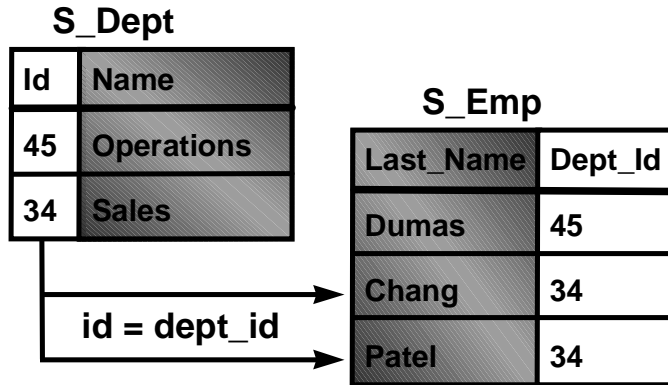
In the report opposite, the G_JOB group appears to the right of the hierarchical model of G_DEPT and its related child group G_EMP; therefore, the department and employee records are formatted in the output before the job title records.

The output shows a master-detail listing of departments and employees. After all departments have been processed, a tabular list of jobs appears.

If you want to output the job information first, move Q_JOB and G_JOB to the left of, or below, G_DEPT. G_EMP is always related directly to G_DEPT because the two groups are owned by the same query.

Using Data Links

- **Equi-join**



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Using Data Links

- **Compound join**
 - Multi-column primary or foreign key
 - Define multiple equi-joins
- **Nonequi-join**
 - Primary key value between two other values
 - Create two links:

```
s_emp.salary >= salgrade.losal
s_emp.salary <= salgrade.hisal
```

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Using Data Links to Structure Data

Introduction

A data link defines a parent/child relationship between a group and a query by way of a *primary key* and *foreign key*. The parent/child relationship causes the child query to be executed for each instance (record) of the parent group, retrieving only rows that match the conditions specified in the link.

Primary and Foreign Keys

A data link usually corresponds to a primary/foreign key relationship between the underlying database tables. The common SQL term for this is a *join*.

Defining Equi-Joins

Create a single link to define an *equi-join*.

An equi-join is a join condition where one value is equal to another.

```
...WHERE s_dept.id = s_emp.dept_id
```

Compound Keys

You can define multiple equi-joins between the same group and query, to implement a *compound* primary/foreign key relationship—that is, a key that consists of more than one column.

Defining Nonequi-Joins

Create two links to define a *nonequi-join*.

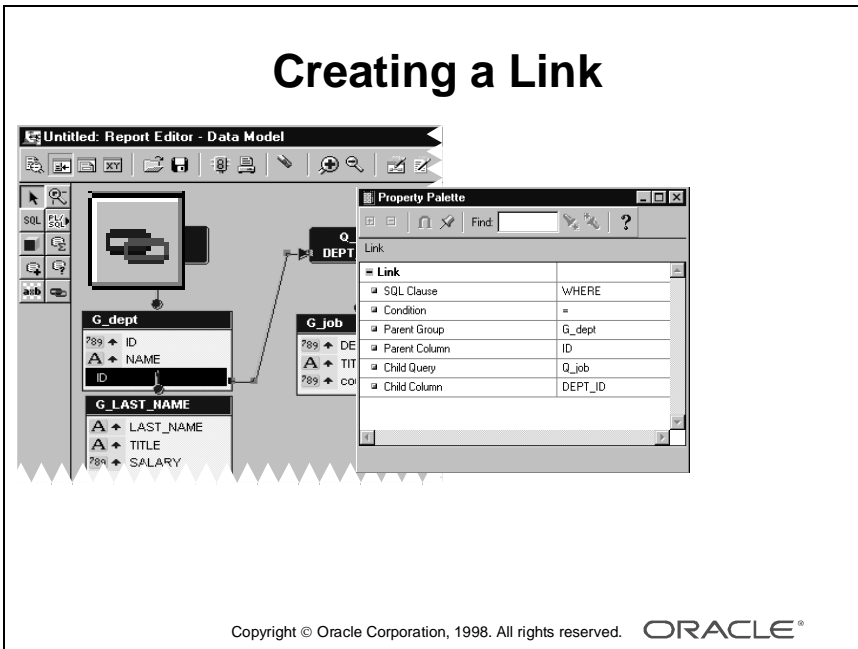
A nonequi-join is a condition where one value falls between a range of two other values.

```
...WHERE s_emp.salary BETWEEN salgrade.losal AND salgrade.hisal
```

In this example, you define links for two separate expressions:

```
s_emp.salary >= salgrade.losal  
s_emp.salary <= salgrade.hisal
```

Creating a Link



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

How to Create a Data Link

Use the Link tool to create a link between groups.

- 1 In the Data Model editor select the Link tool.
- 2 Click the primary key column of the parent group and drag a link to the foreign key column of the child query.

Note: Always create a link from the parent query to the child query.

Modifying the Link Definition

Open the property palette using the right mouse button or double-click the link object in the Data Model or Object Navigator. You can modify the SQL Clause and Condition properties; both have lists of values:

- Change the SQL Clause property from the default WHERE, to HAVING or START WITH.
- Change the Condition to any logical operator in the list, such as >= or <=.

The other properties are read-only. If you want to change the parent or child properties, you must delete and recreate the link.

A Multiple-Query Linked Report

Assume the following two queries within a report.

```
Query: Q_EMP  SELECTd.id, d.name, e.last_name, e.title, e.salary
                FROM      s_emp e, s_dept d
                WHERE     e.dept_id = d.id
```

```
Query: Q_JOB  SELECTdept_id, title, count(id) count_names
                FROM      s_emp
                GROUP BY  dept_id, title
```

You need to link them to display job information for each department, instead of one separate list at the end. The link in the Data Model is *from* Q_EMP:ID column *to* G_JOB:DEPT_ID column.

Other Data Link Methods

Other methods of forming a link:

- **Manual: Using a WHERE clause to form the link (SQL Queries + column objects)**
- **Manual: Using parameters to form the link (Ref Cursor Queries)**
- **Automatic: Using primary/foreign key constraints to form the link (SQL Queries)**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Other Data Link Methods

There are two other types of data link that you can create:

- Manual link
- Automatic link

Creating a Manual Link

A manual link is a link between two group objects, not between specific columns. The link does not define a join condition.

- 1 Select the Link tool.
- 2 Click on the parent group and drag it to the child group.
- 3 Edit the child query to include your own join condition, if applicable.

Note: Since the manual link does not define a join between specific columns, you cannot modify the link definition in the property palette.

Example Assume a simple example of a master query Q_DEPT and child query Q_EMP. In Q_EMP add the join condition:

```
...WHERE s_emp.dept_id = :id
```

Note: “:id” is a bind reference to the primary key column in the master query, Q_DEPT.

Why Create a Manual Link?

Create a manual link for the following reasons:

- Define a complex join condition in a query.
- Implement a cartesian product, linking two tables without a join condition.
- Link a query that contains objects columns or REF columns.
- Link a ref cursor query, using parameters in the Ref Cursor query function.

Creating an Automatic Link

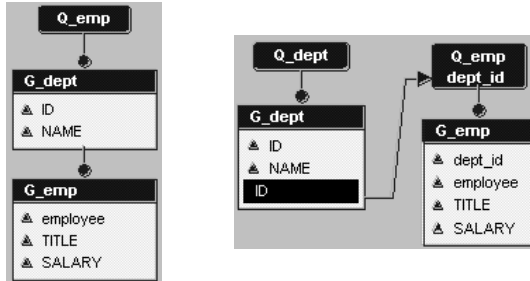
An automatic link is a link between two query objects that automatically defines a join condition if the relevant primary/foreign key constraints exist in the database for columns in the queries.

- 1 Select the Link tool.
- 2 Click on the parent query and drag it to the child query.

An error appears if no relevant constraints exist that match the columns in the master and child queries. You must include the primary/foreign key columns in your queries.

Efficiency Implication of Multiple or Single-Query Hierarchy

- Outer joins
- Data ratio
- Client-Server environment



Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Efficiency Issues: Multiple- or Single-Query Hierarchy

Introduction

You can create the department/employee group hierarchy with two queries instead of one. If you link these two queries, the data structure is the same as the single-query example, that is, a two-level, master-detail hierarchy.

Since you can build a similar report structure using a single-query or multiple-query data model, your choice depends partly on your report requirements—that is, whether you require an outer join report, and on the ratio of master to detail records in your data tables.

Outer Joins

Consider whether you want to display all master records, regardless of any existing details (outer join), or only those master records that have related details.

You can add the outer join notation (+) to a single query, but if you require many or complex outer joins, consider using multiple queries that fetch all records in the master query even if there are no records in the detail query.

Data Ratio

Consider the likely ratio of master to detail rows. As a general guide, use a single query for many master records with relatively few details.

If your data contains relatively few masters with many details each, however, consider using two queries with a link. This means that you retrieve the columns for each master record only once (in the master query) instead of repeating the master columns for every detail row in a single query.

Client-Server Environment

This mode of access is a strong argument for reducing the number of queries on the database to a minimum. Multiple queries for a hierarchical report result in far greater network traffic, in order to fetch records from each query separately.

Note: You can use SQL*Trace to analyze your report; for example, see how much time it is taking to retrieve data from the database.

Summary

- **Groups:** structure data in a group hierarchy.
- **Queries:** write efficient SQL. Test in SQL*Plus.
- **External queries:** save a SQL statement in a file and import into a report.
- **Data links:** create a relationship between data in different queries, normally corresponding to primary and foreign keys.
- **Efficiency:** plan the queries required.

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Summary

Queries

A report performs well only if the SQL code that accesses the database is efficient. Write well-constructed SELECT statements. Use SQL*Trace for detailed analysis of the data access.

You can restrict the rows returned by a query using the Maximum Rows property. This is useful for testing purposes.

External Queries

Create a query in SQL*Plus and test for accuracy and efficiency. Save the final query to a .sql file, then import the query to one or more reports.

Create external queries from the Report Builder, which allows you to use the Query Builder and to save the query in the database instead of a file.

Groups

Organize data into sets of records, allowing you to establish hierarchical reports to any depth and width.

Data Links

Create a data link between two queries, either manually or automatically.

Multiple-Query Reports

A report may comprise many queries running consecutively, or linked together to form a hierarchy. Links are formed according to primary and foreign key relationships.

Efficiency

Consider outer joins, data ratios, and network traffic when designing your reports and planning the number of queries.

Practice 8 Overview

- **Creating a second query in an existing report**
- **Restricting records in a query**
- **Restricting records with a packaged filter**
- **Creating a data link**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Practice 8 Overview

- **Creating a report containing a master group with two details**
- **Creating and referencing an external query file**
- **Creating a report containing a ref cursor query**

Copyright © Oracle Corporation, 1998. All rights reserved. ORACLE®

Practice Session Overview: Lesson 8

This practice session contains:

- Creating a second query in an existing report
- Restricting records in a query
- Restricting records with a packaged filter
- Creating a data link
- Creating a report containing a master group with two detail groups
- Creating an external query file and referencing the file in a new report
- Creating a report containing a ref cursor query

Introduction

In this practice session you include new data in a report by adding a query to an existing single-query report created with the Report Wizard. You also practice linking data from different queries to alter the report structure. Using two different techniques, you restrict the records in a report. If you have time, you create an external query and link it to a report.

Practice Session: Lesson 8

1 Open report `p8q1_a.rdf`, which contains a single-query report showing customer information.

a In the Data Model create a new query. Either use the Query Builder or import filename `p8q1.sql`. Change the group name to `G_ORD`.

```
QUERY : Q_ORD
SELECT customer_id, id, date_shipped, total
FROM s_ord
```

b Open the Report Wizard and choose a Tabular layout.

c Copy group `G_ORD` to the Displayed Groups area.

d Copy all remaining fields to the Displayed Fields area.

e Choose Finish.

f Note how the data appears in the Live Previewer. The `Q_CUSTOMER` data appears before the `Q_ORD` data. Remember that you can alter the order in which the data is printed by modifying query order in the Data Model. You may want to try that.

Note: If you do adjust the order of the queries, put them back in the original order and save before the next question.

g Save the report as `p8q1.rdf`. Do not close the report.

2 Modify the previous report to restrict the number of customer records processed by the group.

a Place a filter on the `G_CUSTOMER` group to display only the first four customers. Run to test.

b Save the report as `p8q2.rdf` and close it.

c Close the report, and any other you still have open in the Object Navigator.

3 Open `p8q1.rdf`.

a In the Data Model link the two queries together on the primary and foreign keys.

Hint: To create the link, select the Link tool. Click and drag from the primary key field in the Customer group to the foreign key field in the Ord group.

b Redo the layout by opening the Report Wizard. Choose Group Above style. Do not display `CUSTOMER_ID`.

- c Run to test.
 - d Save the report as `p8q3.rdf`.
- 4 Modify `p8q3.rdf` to show and restrict customers' credit rating. You need to modify the query properties.
- a Include `CREDIT_RATING` in the query statement, although you need not display it. Restrict the query statement to show those who have a poor credit rating.
 - b Run to test.
 - c Save the report as `p8q4.rdf` and close it.
- 5 Create a new report that lists the sales representatives and the products they have sold.
- a Choose a Group Above report style.
 - b Import the query file `p8q5_a.sql`.
 - c Make `NAME` and `ID` the group fields. Make sure that these fields are in one group, at Level 1, not two separate groups.
 - d Do not display `ID` but display all other fields in the order described in the table below.
Hint: You can change the order of fields in the Report Wizard on the fields tab.
 - e Change the following field attributes:

Column	Label	Width
name	Name	20
product	Product	10
product_id	ID	4
quantity	Quantity	4
prod_total	Total	6

- f Use the template file `submit.tdf`.
- g Run and test your report.

- h** Now you need to show not only which products the sales representatives have sold but also who are the customers they have done business with.
Go to the Data Model and add a new query, importing the file `p8q5_b.sql`. This query returns the customer names, their IDs, and the total sales per customer for each sales representative.
- i** Rename the group to `G_CUSTOMER`.
- j** Link the second query to the correct group.
- k** Use the Report Wizard to display the new group. Do not display `SALES_REP_ID` or `ID`.
- l** Change the following field attributes:

Column	Label	Width
customer	Customer	10
customer_id	ID	4
cust_total	Total	6

- m** Run and test your report.
- n** Save the report as `p8q5.rdf` and close it.

If You Have Time

- 6** Create an External Query module that selects the following data:

Columns	Table
id, name	s_dept
id, first_name ' ' last_name, title, salary	s_emp

- a** Remember to include a join condition.
 - b** Give the columns meaningful names.
 - c** Save the query as `p8q6.sql` and close the query window.
- 7** Use the Report Wizard to create a Group Left report using the external query created in question 6.
 - a** Department information should be in the master group.
 - b** Look at the Data Model. Notice that the break group exists.

- c In the Data Model, move TITLE into a group at the top of the hierarchy.
Hint: Move it into the department group first, then up into its own new group.
 - Note:** Since this is a single-query report, you can also make the group change using the Report Wizard Group tab. However, you have learned that you can combine many queries and groups in your own reports, and this simple exercise helps you to practice moving columns in the data model.
 - d Open the Report Wizard and choose Finish to update the layout and run the report.
 - e Run and test the report.
 - f Save the report as `p8q7.rdf` and close it.
- 8 Create a new report, using a ref cursor query.
- a In the Object Navigator, create a package defining a record and a ref cursor. The record has the following fields:

Name	Definition
last_name	s_emp.last_name%type
first_name	s_emp.first_name%type
salary	s_emp.salary%type

- Note:** You can import the file `p8q8a.txt`.
 - b In the Data Model create a ref cursor query. The function opens the cursor for the following select statement:

```
SELECT last_name, first_name, salary
FROM s_emp
```
 - Note:** You can import the file `p8q8b.txt`.
 - c Open the Report Wizard to create the layout for the report. Select all the fields and choose a template.
 - d Run and test the report.
 - e Save the report as `p8q8.rdf` and close it.
- Do you still have any objects open in the Object Navigator? If you do, close them all.

**Enhancing Reports Using
the Data Model: Creating
Columns**

Objectives

At the end of this lesson, you should be able to do the following:

- **Describe various Data Model columns**
- **Display the contents of a file**
- **Identify characteristics of user-defined columns**
- **Create report summaries and subtotals**
- **Derive data, using a formula column**
- **Create and populate a placeholder**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

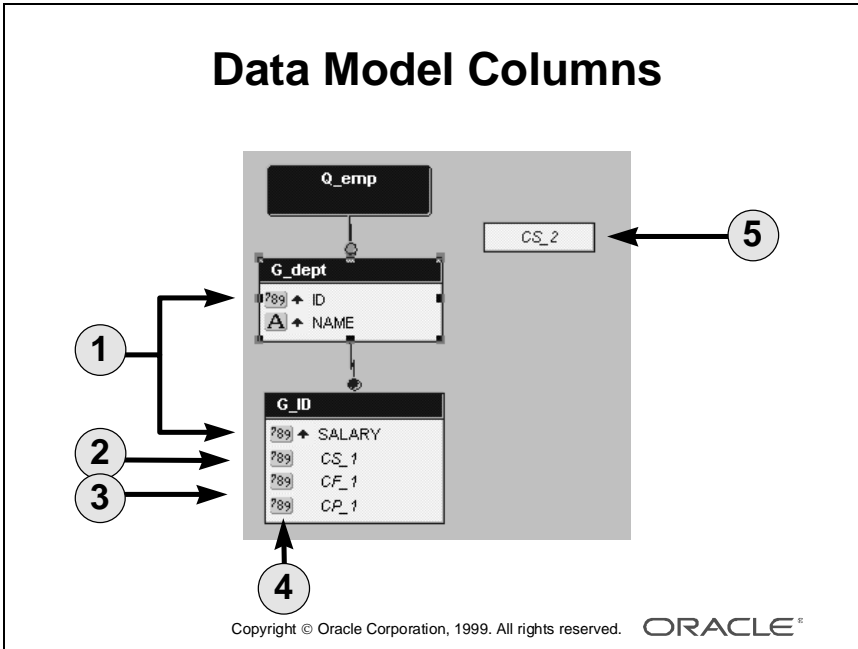
When you define a query to extract the data from the database and modify the group structure, you might want to modify the data model further, to perform complex calculations not included in your query.

In this lesson you learn more about the Data Model objects and how to enhance reports by creating user-defined columns for summaries and calculations.

Objectives

At the end of this lesson, you should be able to do the following:

- Describe the different types of Data Model columns
- Display the contents of a file, using a database column
- Identify the characteristics of the three types of user-defined columns
- Create report summaries and subtotals, using summary columns
- Derive data using a formula column
- Create and populate a placeholder column



1	Database columns	4	Placeholder column
2	Summary column	5	Summary column at report level
3	Formula column		

The Data Model Column Types

Introduction

A Data Model column is a data container. A column defines the type of data, but does not include any formatting information relating to the report output.

The Column Types

There are two main types of column: those that Report Builder creates from your query and those that you create yourself.

- Database columns: Report Builder creates one column for each item in the SELECT statement of a query. Database columns are directly associated with a column or expression retrieved from the database by the query. You cannot modify most properties of a database column.
- Developer-created columns: You can create columns, either in a specific group or at report level.

Developer-Created Columns

A developer-created column may be one of three types:

- Summary column: Summarizes another column and can recalculate for each record in a specified group.
- Formula column: Uses a formula you have written in PL/SQL to calculate a value from one or more other columns.
- Placeholder column: Has its value set from another object at run time.

Default Column Names

Each column that you create in the Data Model has an initial default name, which you should change to a descriptive name.

Column Type	Default Name
Summary	CS_1
Formula	CF_1
Placeholder	CP_1

Maintaining Database Columns

Database Column: SALARY	
General Information	
Name	SALARY
Comments	
Column	
Column Type	Database - Scalar
Datatype	Number
Width	11,2
Value if Null	
Break Order	Ascending

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**[®]

1	Read-only properties
---	----------------------

Maintaining Database Columns

Introduction

Because Report Builder creates database columns from the contents of the query statement, there are some restrictions to how you modify these columns.

Displaying a Column Property Palette

To display a column's property palette, as for other report objects, you can use one of several methods:

- Double-click on the Column icon in the Object Navigator.
- Double-click on the column in the Data Model.
- Select the column and use the right mouse menu.
- Select the column and use the Tool menu.

Editing Database Column Properties

For database columns, some of the property settings are not editable in the property palette. You can see the values, but you cannot modify them. These values are dependent on the column name and type in the SQL query.

The read-only properties are:

- Name
- Column Type
- Datatype
- Width

All editable properties, except Value if Null, provide a list of values.

Some properties may or may not appear in the palette, depending on the column datatype, or on the value of another property.

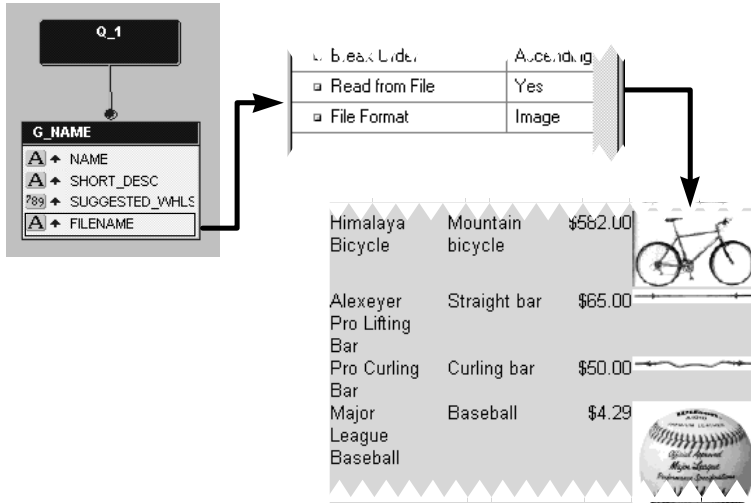
Note: The column types Database - Object and Database - Ref refer to object columns and reference columns in Oracle8. Their properties should not be changed.

For further information, see Appendix F.

Deleting a Database Column

You cannot delete a database column object directly from the group. To delete a column, you must remove the corresponding expression from the SELECT statement in the query.

Outputting Contents of a File



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Using a Database Column to Output Contents of a File

Introduction

You can include in your report the contents of a file, for which the file name varies for each record retrieved. The file can contain an image—for example, the picture of each employee or of each product, as shown opposite.

What Is a File Column?

A *file column* is a database column that contains the name of an operating system file or a URL. The file can be text, image, or multimedia.

The Read from File property does not exist in the property palette unless the column is of the type Database - Scalar and has a datatype of Character.

What File Formats Can You Output?

Report Builder provides a list of values for the file type: Text, Image, CGM, Oracle Format (drawing), Sound, Video, OLE2, and Image URL.

Note: Text is the only valid format in character mode reports.

How to Create a File Column

- 1 Open the column property palette.
- 2 Set Read from File to Yes.
- 3 Select the appropriate file format from the list of valid types.

What Does the Report Wizard Create?

- Video or Sound format: Button
- All other file formats: Field

The Report wizard provides a default size for the displayed field. However, you can resize it either in the Live Previewer or the Layout Model editor.

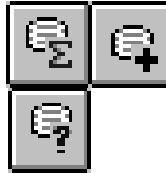
File Search

For portability, do not prefix the full path to the filename in the column. If you do not prefix a path, Report Builder can use its file path search order to find the file.

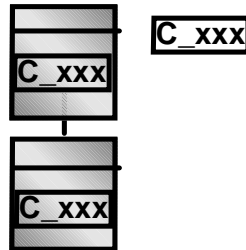
Include the correct paths in the environment variable `REPORTS60_PATH`.

Creating a Column

- **What type of value?**
Choose the correct column tool



- **What frequency?**
Create in a group or at report level



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating a Column

Introduction

There are two important questions to ask yourself before you create a column:

- What type of column do I require? The answer determines which one of the three tools to select.
- How often do I want this column to be displayed?—The frequency at which the column value appears in the output determines the group in which you create the column.

How to Create a Column

- 1 Select the appropriate tool from the toolbar in the Data Model editor. Click inside a group to create the column at the same frequency as other columns in that group.
Note: To display the column once for the report, click in an open area of the Data Model editor.
- 2 Double-click on the column to display its property palette.
- 3 Replace the default column number with a brief but descriptive name. Consider standard naming conventions for different types of column.
- 4 Fill in the rest of the settings as appropriate for the type of column you require.

Summary Column

- **Specific properties:**
 - **Function**
 - **Source**
 - **Reset At**
 - **Compute At**
- **Datatype depends on Source datatype**
- **Page summaries: not supported in the wizard**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Summary Columns

Introduction

A summary column performs a computation on another column's data.

Summary Column Properties

The following properties apply specifically to summary columns:

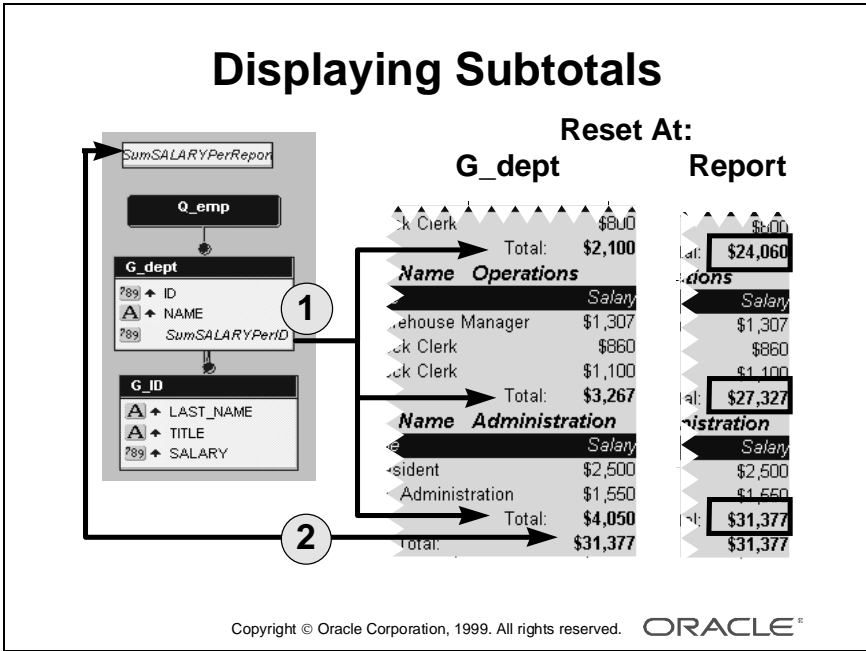
- **Function:** The calculation to be performed on the values of the column specified in Source.
- **Source:** The name of the column whose values are to be summarized. Source is a list of values containing those columns that are valid for the specified Function.
- **Reset At:** The group at which the summary column value resets to zero.
- **Compute At:** The group for which a % of Total summary column is computed. Compute At is used only for columns with a function of % of Total.

Summary Datatypes

The datatype of a summary column depends on the datatype of the source of the summary. If you change the datatype of the source column, the datatype of the summary also changes.

Page Summaries

The Report wizard does not support page summaries. If you select a page summary in the Field tab of the Report wizard, an error message appears. Deselect the page summary from the Field tab and create the field manually in the Layout editor. Creating fields in the Layout editor is described later in the course.



1	Department totals
2	Report total

Example Report Displaying Subtotals

Introduction

The control break report output opposite shows two summary columns created by the wizard. One summary shows the total salary for each department, the other shows the total salary for the whole report.

The wizard creates these summaries at one or more levels above the salary source column.

Resetting Summary Values

The Reset At property defines the group at which a summary resets to zero to allow you to recalculate a number of subtotals.

The wizard chooses a default reset value for these summaries.

- The department total, in G_Dept, is reset at G_Dept—that is, values start from zero for each new department record.
- The report total, at report level, is reset at Report.

To create an accumulating total for the department total, change the Reset At property to Report.

Valid Reset Options

The Reset At property displays only options that are valid for each summary.

There are two options, Report and Page, that are valid for all summaries. Depending on the level of the summary, some groups are also available in the valid list.

Outputting Summary Values

The frequency at which the summary value is displayed depends on the group within which you create the column.

Rule

To create your own summaries, use the following rule:

Discrete Total: Reset At value is the same as the summary group.

Accumulating Total: Reset At value is a higher group than the summary.

Displaying Percentages

Q_emp

G_dept

- ⌘ ID
- ⌘ NAME
- ⌘ Total/SALARYPerId

G_ID

- ⌘ LAST_NAME
- ⌘ TITLE
- ⌘ SALARY
- ⌘ Total/SALARYPerLAS

Reset At: **G_emp**
Compute At: **G_dept**

G_emp Report

Salary	% of Total	Salary	% of Total
\$800	38.10%	\$800	2.55%
Operations			
\$1,307	40.01%	\$1,307	4.17%
\$860	26.32%	\$860	2.74%
\$1,100	33.67%	\$1,100	3.51%
Total: 10.41%		Total: 4.1%	
Administration			
\$2,500	61.73%	\$2,500	7.97%
\$1,550	38.27%	\$1,550	4.94%
Total: 12.91%		Total: 12.91%	

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

1	Department total percentage
2	Employee total percentage

Example Report Displaying Percentages

Introduction

The control break report opposite shows the following two percentage totals created by the wizard:

- Employee's salary as an individual percentage
Notice that this is a column in the G_Emp group. This is the only summary function for which the wizard creates a column in the same group as its source.
- Department total as a percentage of the report total

Resetting Percentages

The Reset At property works the same way that the Sum function does. The default, Reset At G_Emp, results in a discrete value for each employee.

For both summaries opposite, the Reset At property is G_EMP. Both percentages reset to zero for each employee record. If you want to create a running percentage that eventually accumulates to 100%, change the Reset At property to a reset at a higher level than its source column. For example, to create an accumulating percentage that accumulates for each department and then resets to zero, change Reset At to G_Dept.

The Compute At Property

The Compute At setting defines the total value that a % of Total summary uses in the percentage calculation. This property is only applicable to % of Total summaries.

The wizard sets the Compute At property to one group above the source column. In the left example opposite, the default is G_Dept. To display the salary as a percentage of the whole report value, change Compute At to Report.

Resetting Summary Values

Data Model Group	Reset At		
	REPORT	G_DEPT	G_EMP
REPORT	Grand Total	XXXX	XXXX
G_DEPT	Running Total	Sub Total	XXXX
G_EMP	Running Total	Running Total	Record Total

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Resetting Summary Values

The Data Model group defines how often a value is calculated. Therefore Report Builder does not allow you to reset a calculated value at a lower level of the hierarchy; that is, you cannot reset a value more often than you calculate it.

Valid Group and Reset Options

The table on the opposite page identifies which combinations of group and reset options are valid. The XXXX denotes invalid combinations.

Formula Column

- Performs a user-defined computation
- Executes a PL/SQL function
- Must return a value
- Can be Character, Number, or Date
- Returned value must match datatype

```
function CF_SALCALCFormula return Number is
begin
    return(my_function(:salary));
end;
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Formula Columns

Introduction

A formula column performs a user-defined computation on the data of one or more other columns data.

The PL/SQL Formula property button opens the Program Unit editor where you enter and edit your code.

Code in a Formula Column

The formula is a PL/SQL function that returns a single value. It must return a value of the same type as specified in the datatype property. The following code is a simple example that calls another function, passing the :salary value as an argument, and returns the result in the SALCOMM formula column.

```
function salcomm return number is
begin
    return(my_function (:salary));
end;
```

Rule

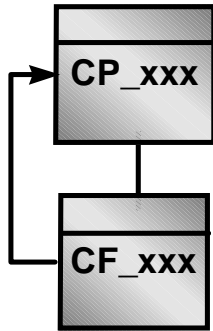
Any columns referenced by the formula column are calculated first.

A Note on Efficiency

The code above is a simple example to illustrate the syntax. The function “my_function” might be a local program unit at report level, a program unit in an attached PL/SQL library, or a stored program unit.

Consider using a formula column only if you cannot include your calculation in the query statement. It is always advisable to perform calculations on the server rather than on the client, especially if the calculation requires only database values.

Placeholder Column



- An empty container at design time
- Populated by another object at run time
 - Before Report trigger
 - Formula column at report level
 - Formula column in same group or below placeholder

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Placeholder Columns

Introduction

A placeholder column is an empty container at design time. The placeholder can hold a value at run time that has been calculated and placed into it by PL/SQL code from another object.

Setting the Value of a Placeholder Column

You can set the value of a placeholder column in the following places:

- A Before Report trigger
- A report-level formula column (if the placeholder column is at report level)
- A formula column in the placeholder group or a group below it

Uses of Placeholder Columns



- Placeholder columns enable you to populate multiple columns from one piece of code. You can calculate several values in one block of PL/SQL code in a formula column and assign each value to a different placeholder column. You therefore create and maintain only one program unit instead of many.
- Store a temporary value for future reference. For example, store the current maximum salary as records are retrieved.

The Placeholder's PL/SQL Formula Property

You cannot populate a placeholder by writing code in the placeholder's own Formula property.

The PL/SQL Formula button opens up the Program Unit editor where you enter and edit your code. This is applicable only when you call a user exit.

Populating a Placeholder

Maximum Salary: \$2,500

Maximum Earner: Velasquez in Department 50--Administra

Id 42

Name Operations

Last Name	Title	Salary
Menchu	Warehouse Manager	\$1,250
Patel	Stock Clerk	\$795
Nozaki	Stock Clerk	\$1,200

Id 43

Name Operations

Last Name	Title	Salary
Eiri	Warehouse Manager	\$1,100
Markarian	Stock Clerk	\$850
Newman	Stock Clerk	\$750

Id 44

Name Operations

Last Name	Title	Salary
Catchpole	Warehouse Manager	\$1,300

Q_emp

G_dept

? ID

A NAME


G_ID

A LAST_NAME

A TITLE

? SALARY

? CF_CALC_MAX

Copyright © Oracle Corporation, 1999. All rights reserved. 

Example Report to Populate a Placeholder

Introduction

The sample report opposite is a report of all employees. The aim of the report is to:

- Calculate and temporarily store the name of the employee who earns the highest salary in the company
- Display the highest earner and the maximum salary once at the beginning of the report

The Columns

You need to create the following columns:

- A summary to show the maximum salary for the company
- A placeholder to contain the highest earner's name at runtime
- A formula to:
 - Compare each employee salary with the maximum salary
 - Populate the placeholder with the employee name if salary equals maximum salary

How to Create the Columns

- 1 Create a summary column CS_MAX_SALARY at report level (outside all groups). Datatype: Number; Width: 10; Function: Maximum.
- 2 Create a placeholder column CP_MAX_EARNER at report level (outside all groups). Datatype: Character; Width: 80.
- 3 Create a formula column CF_CALC_MAX in the G_EMP group. Datatype: Number; Width: 10 (these are the default values).
- 4 Open the Program Unit editor from the PL/SQL Formula property in CF_CALC_MAX. Create a function:

```
function CF_CALC_MAXFormula return Number is
begin
    if :salary = :CS_MAX_SALARY then
        :CP_MAX_EARNER := (:LAST_NAME||' in Department '||
                           :ID||'--'||:NAME);
    end if;
return(0);
end;
```

Summary

- **Column types**
- **Summary levels**
- **Reset At—resets to zero**
- **Compute At—% of Total only**
- **Formula columns return a value**
- **Placeholder columns hold a value**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE[®]

Summary

Column Types

- A database column exists for each column or expression in the query
- Create summary columns for computations and summaries
- Create formula columns to perform a calculation or to populate a placeholder column
- Create a placeholder column to provide an empty container that is populated from another object at run time

Summary Levels

- Report
- Page (not supported by the Report wizard)
- Groups within the Report

Reset At

The level at which to reset the summary back to zero

Compute At

Valid only for summary columns with a function of % Total

Practice 9 Overview

- **Creating a new report with summaries**
- **Adding summary calculations to an existing report**
- **Creating a report that displays the contents of a file in a column**
- **Creating a new report with ranking summary columns**
- **Adding placeholders for highest and lowest values**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 9

This practice session contains:

- Creating a new report with summaries
- Opening an existing report to add summary calculations to the columns
- Creating a report that displays the contents of a file in a column
- Creating a new report with ranking summary columns
(Use ORDER BY clause to create the ranking effect.)
- Opening an existing report to add placeholders for the highest and lowest values in the report

Introduction

In your reports, you may need to include additional columns with summary functions, or even add summaries, such as break totals and report totals. This practice gives you the opportunity to create summaries and additional columns with functionality that cannot be achieved using SQL alone.

Practice Session: Lesson 9

1 Create a new report using the Report wizard. Choose a Group Above report style.

a Display the following data:

Table	Column	Definition	Group	Label	Width
s_ord	id		G_ord	Ord. No.	4
	date_ordered			Order Date	9
s_item	item_id		G_item	No.	2
	product_id			Product No.	5
	price			Price	4
	quantity_shipped			Quantity	4
	line_total	price*quantity_shipped		Total	4

b Do you need a formula column to calculate the line_total?

c Create the following summaries. Can you use the Report wizard to create the summaries?

Column	Summary
line_total	Sum
line_total	% Total

d Use the `summit.tdf` template file from the file system.

e In the Data Model, rename the groups to G_ORD for the break group and G_ITEM for the detail group. Alter the position of columns in the G_ORD group so that DATE_ORDERED is above ID, and change the break order on DATE_ORDERED so that it is sorted in descending order.

f Run and test the report. Do not reenter the wizard.

g Save the report as `p9q1.rdf` and close it.

- 2 Open report `p8q5.rdf`.
- Add a summary to total `PROD_TOTAL` for each employee master. Can you use the Report wizard?
 - Create columns to show the percentage of the total per sales representative for `PROD_TOTAL` and `CUST_TOTAL`.
 - Make all the new summaries four characters wide.
 - Run and test the report.
 - Save the report as `p9q2.rdf` and close it.
- 3 Create a new report using the Report wizard.
- Choose a Tabular report style and display the following information:

Table	Column	Label
<code>s_product</code>	<code>id</code>	ID
	<code>name</code>	Name
	<code>short_desc</code>	Description
<code>s_image</code>	<code>filename</code>	Picture

- Restrict the query to return only products with ID beginning with 10 or 30, because you do not have picture files for all products.
- Run the report.
- In the Data Model change the properties for the column `filename` to read from an image file.
- Use the Report wizard to redo the layout.
- In the Live Previewer resize the picture column to about 3x3 inches. **Hint:** You need to enable Flex Mode. Use the last image on the page to resize the picture column.
- Save the report as `p9q3.rdf` and close it.

- 4 Open report `p9q1.rdf` to list the orders ranked by their total.
 - a In the Data Model add column `s_ord.total` to the query.
Hint: You have to move the new column `TOTAL` to the `G_ORD` group.
 - b Because you want to list the highest total first, set the break order for `TOTAL` accordingly, and move the `TOTAL` column so that it is first in the `G_ORD` group.
 - c To display the ranking, add another column called `RANK` in the group `G_ORD`.
 - d Use the Report wizard to display the rank, giving it a width of 2. You need not display the total, because there is already a total on the `G_ITEM` group.
 - e Run the report to test.
 - f Save the report as `p9q4.rdf` and close it.

If You Have Time

- 5 Open report `p9q1.rdf` to display the highest and lowest orders at the end of the report.
 - a In the Data Model add the columns necessary to list the maximum order total and the corresponding ID as well as the minimum order total and the corresponding ID.
Hint: You will need one formula column and some summaries and placeholders.
You can import the code for the formula column from the file `p9q5.txt`.
 - b Use the Report wizard to put the new columns into the layout. Display the maximum order total, the maximum ID, the minimum order total, and the minimum ID.
 - c Run the report to test. Scroll down to the last page to see the new columns.
 - d Save the report as `p9q5.rdf` and close it.

**Enhancing Reports Using
the Layout Model**

Objectives

At the end of this lesson, you should be able to do the following:

- **Distinguish the report regions and sections**
- **Design multipanel reports**
- **Describe the layout objects and relationships**
- **Modify an existing report layout**
- **Create variable length lines**
- **Create buttons and explicit anchors**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

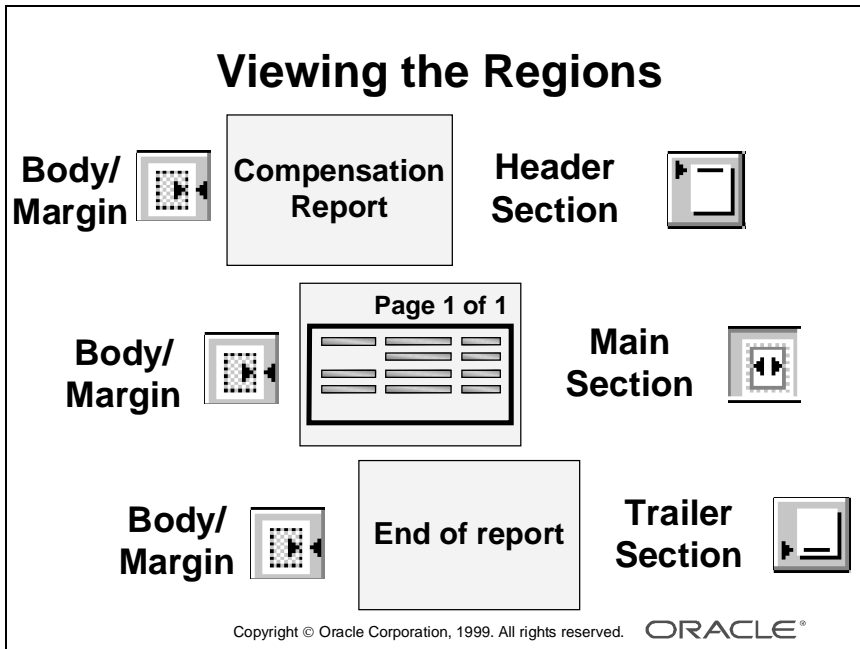
You can fulfill many of your report requirements simply by making modifications in the Live Previewer. However, for more complex reports you need to modify or insert objects in the Layout Model editor.

The layout can be a complex area to view and modify. This lesson aims to give you an appreciation of the layout regions, as well as the objects that you see and create using object tools in the toolbar.

Objectives

At the end of this lesson, you should be able to do the following:

- Distinguish the different regions and sections of a report
- Design multipanel reports
- Describe the layout objects and their relationship
- Modify an existing report layout using the Layout Model tools
- Separate report data using variable length lines
- Add interactivity to a report using a button
- Alter object positions at run time, using an explicit anchor



Viewing the Layout Model Regions

Introduction

The Layout Model editor allows you to modify one of two different report regions in one of three sections at any time. The regions separate the formatting of margins from the body of your report pages in any section.

When you enter the Layout Model editor, you see the body region of the main section of your report. To change the region in the editor, choose the Edit Margin tool or choose View—>Layout Section—>Edit Margin.

Each physical page consists of a body and margin region.

Body

The body region of a section can contain text, graphics, data, computations, and summaries.

Margin

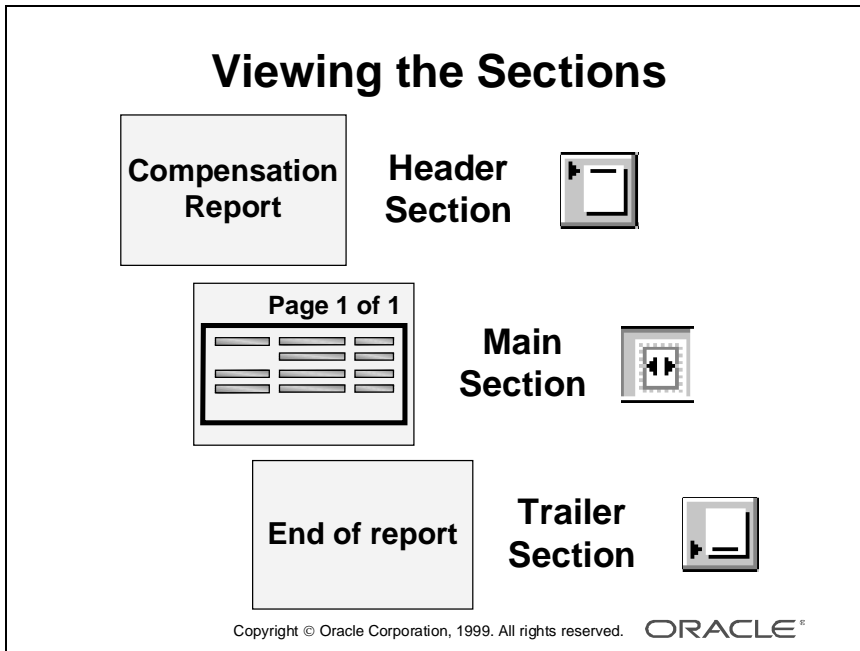
Margins can contain text, graphics, page numbers, page totals, and grand totals.

When you choose the margin view, the body also remains visible, but read-only, in the editor. The body appears within a black rectangle that defines the size of margin and body. You cannot modify objects in the body while working in the margin view.

Adjusting Margin Size

To adjust the size of the margin, choose the margin view, select the margin rectangle, click on a select handle, and drag to resize.

Modify margins in your template definition to apply the same dimensions to many reports.



Viewing the Report Sections in the Layout Model

Introduction

The Layout Model editor allows you to modify the regions in one of three sections at any time. Each section has its own body and margin regions. This allows you to define multiple layouts for the same data model, producing output in a number of styles.

When you enter the Layout Model editor, you see the body region of the main section of your report. To change the section, choose the Main Section, Header Section, or Trailer Section tool, or choose View—>Layout Section—>Main Section, Header Section, or Trailer Section.

Header Section

The header section consists of one or more pages that appear once, on a separate set of pages, as the first part of the report.

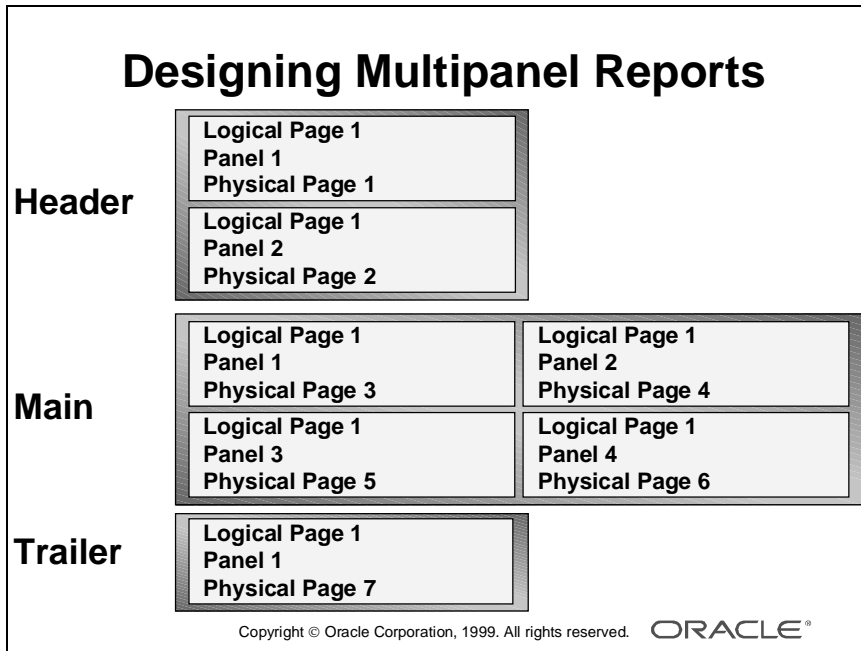
Main Section

By default, the Report Wizard creates the layout in the Main Section of a report, positioned between the header and trailer pages.

Trailer Section

The trailer section consists of one or more pages that appear once, on a separate set of pages, as the last part of the report.

Note: The names of the sections are Header, Main, and Trailer Section for compatibility with prior releases.



Designing Multipanel Reports

Introduction

Report Builder allows you to design large, or multipanel, report layouts by maintaining two different definitions of a page.

- A *physical page* represents the actual paper page size.
- A *logical page* represents a conceptual page on which you can design the layout of your report.

Logical Panels and Physical Pages

Report Builder allows you to design multipanel reports where the logical design of the report output is wider than the physical (printer) page. A report might need a layout width of 16 inches that can be printed on two physical printer pages each 8 inches wide.

The unit of measurement is a property of the report object itself.

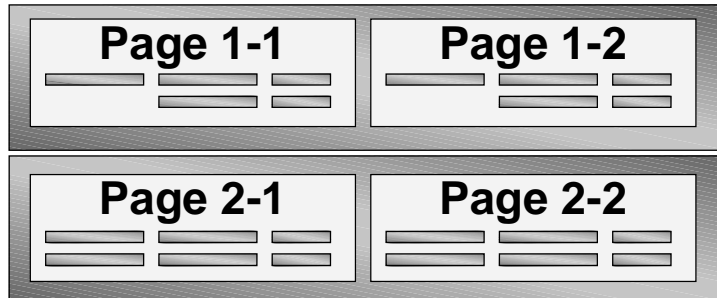
How to Define Dimensions or Trailer Section Node in the Object of the Report

- 1 Select the Header Section, Main Section, or Trailer Section node in the Object Navigator.
- 2 Open the property palette.
- 3 Specify physical page size: Report Width and Height properties.
- 4 Specify logical page size: Logical Horizontal/Vertical Panels properties. Define the logical dimensions in multiples of the physical page width and height.
- 5 Define the page orientation as required.

Example In the diagram opposite, the layout is twice the width of the physical page. For example, to output a logical design of 16 inches across 8-inch pages, the settings are:

- Unit of Measurement: Inch
- Physical Page Height: 8
- Logical Horizontal Panels: 2
- Logical Vertical Panels: 1

Printing Multipanel Reports



Logical horizontal panels = 2

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

How to Print Panels

The default order in which panels are printed is Across/Down. That means that horizontal panels are printed before vertical panels. To change this order, modify the Panel Print Order property of the report object.

- 1 Select the report node in the Object Navigator.
- 2 Open the property palette.
- 3 Specify the Panel Print Order property.

Layers of a Tabular Report



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

The Layout Model Layers

Introduction

All layout objects exist in a hierarchy. Each object is contained inside and on a layer above another object. The hierarchy is very important in the processing of report objects. If you destroy the correct layering, you receive error or warning messages, and the report might not run.

The bottom layer of a report is effectively the page itself.

Layers of a Tabular Report

To explain the concept of layers, the exploded diagram shows the objects in a simple tabular report. From bottom up, the layers are:

- Body
- M_G_xxx: Group frame encloses other objects and controls the format, frequency, and position of several objects simultaneously
- R_G_xxx: Repeating frame is displayed for each row of data that is retrieved for a group
- F_xxx: Fields define the appearance of columns; can contain character, number, and date for each database column

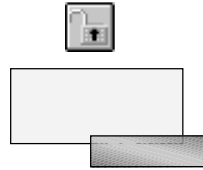
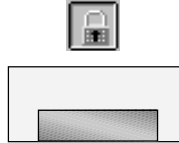
In addition, the tabular style creates a boilerplate text object, B_xxx, for each field label; these objects occur outside the repeating frame, not once for each record.

Group Reports

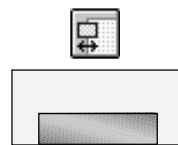
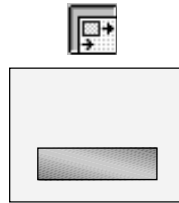
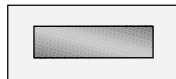
Group Left and Group Above reports create two repeating frames; the detail group frame is nested inside the master repeating frame, to produce a detail tabular listing for each master record.

Avoid Layout Errors

- **Confine Mode**



- **Flex Mode**



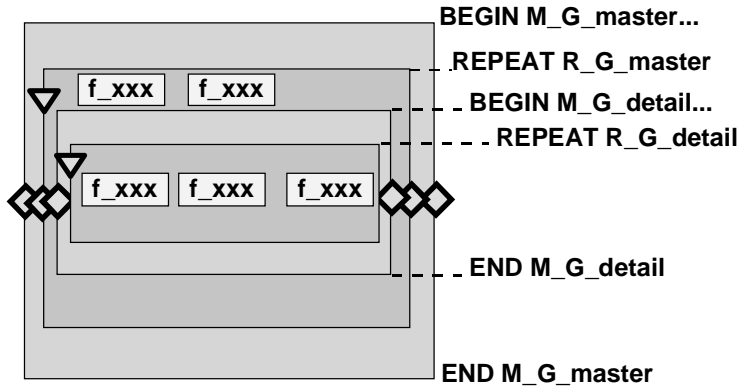
Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Avoiding Layout Errors

To avoid common hierarchy errors in your report layout, always work in the Layout Model using the Confine and Flex modes.

- Confine Mode
 - Work with Confine Mode enabled (closed padlock) to avoid accidentally moving one object outside or below its correct enclosing object.
 - Work with Confine Mode disabled (open padlock) when you want to move one object outside its enclosing object.
- Flex Mode
 - Work with Flex Mode enabled to adjust all affected objects when you move or resize one object; the whole layout flexes to accommodate your changes.
 - Work with Flex Mode disabled when you want to move an individual object without moving other objects.

Report Processing



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Report Processing

To understand how the Layout Model maps to the output that you see in the Live Previewer, consider the layout as a 3GL program, where the group frame is a WHILE loop and the repeating frame is a block.

In a two-level group report, the report processes a master record, including all details that relate to each master record, and then processes the next master.

```
BEGIN M_G_master
while M_G_master has records to process
  REPEAT R_G_master
    BEGIN M_G_detail
    while M_G_detail has records to process
      BEGIN R_G_detail
      end R_G_detail;
    END M_G_detail;
  END R_G_master;
END M_G_master;
```

Layout Symbols

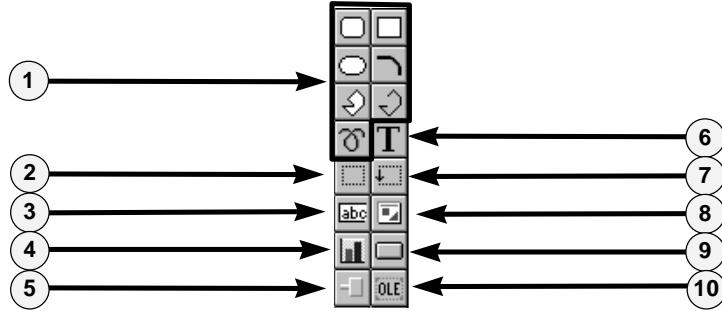
Layout objects often show symbols and icons that indicate their characteristics. The diagram opposite shows two of these symbols:

- Black arrowhead: Indicates a repeating frame object.
- Diamond: Indicates that the object is a variable size; it can increase or decrease in size depending on the contents.

Creating Layout Objects

The tool palette contains:

- Standard GUI drawing tools
- Frame, repeating frame, field objects
- Other layout objects



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Drawing objects	6	Text
2	Frame	7	Repeating frame
3	Field	8	Link file
4	Chart	9	Button
5	Anchor	10	OLE2 object

Creating Layout Objects

Introduction

Layout objects define the appearance of the report output and are displayed in the Layout Model editor.

When you create or modify your report using the Report wizard, Report Builder automatically creates many objects, such as the frame, repeating frame, fields, and labels for a tabular report style.

The tool palette provides tools for you to create these and other types of objects. The tools available depend on the view currently in the Layout Model.

Layout Object Naming Conventions

When you create a layout object, the default name is an alphabetic character that indicates the object type, followed by a number. Rename your layout objects meaningfully to help you in developing and maintaining the report.

Additional Layout Objects

Object Type	Default Name	Description
Chart	D_	A display object that represents data as a graph, such as a pie or bar chart; Oracle Graphics runs the chart and returns the result to the display object in your report
Boilerplate	B_	Constant text or an image that appears in a report every time it is run, and that is not dynamic; example: the Report wizard creates column labels and template objects
Link File	B_	Constant text or an image that appears in a report every time it is run, but is read from a file on disk
OLE2	B_	A layout object that calls an OLE2 registered product to display something in your report
Button	U_	An interface element that can perform a PL/SQL procedure, or display a multimedia file; applicable only in the Previewer
Anchor	None	A layout object that anchors the position of one object relative to another; create an anchor to override the default position

Layout Tools



• Pin a tool: double-click



• Shift for constrained objects



• Magnify



• Frame Select



• Additional Default Layout

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

The Layout Model Tools

Introduction

In addition to the object tools in the tool palette, the Layout Model provides some specific tools to help you modify layout objects.

Pinning a Tool

If you want to create several objects of the same type, double-click the drawing tool. A small red pin appears on the tool icon, and the tool remains active until you click the tool again or choose a different tool.

Drawing Constrained Objects

To create a constrained object, such as a perfect circle or square, hold down the constrain (Shift) key when drawing the object.

To resize an object to scale, hold down the constrain key when resizing the object.

Magnifying the Layout

Choose the magnify tool when you want to focus on one part of a large report. The magnify tool centers the layout area on the point that you choose and enlarges objects to twice their size.

Hold down the constrain key when using the magnify tool to reduce objects by half.

There are also zoom in and zoom out buttons in the toolbar.

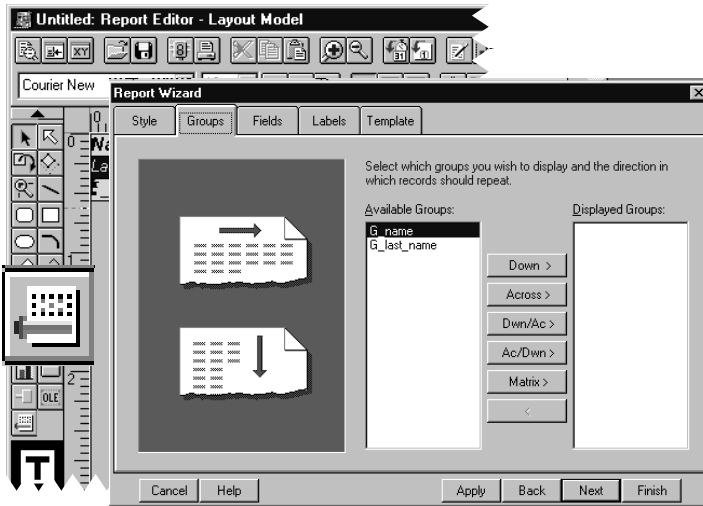
Selecting All Objects in a Frame

Choose the Frame Select tool and then click on a frame or repeating frame in the layout. This selects the frame and all objects inside it.

Additional Default Layout

The Additional Default Layout tool invokes the wizard to enable you to create additional layouts in your report without deleting existing objects.

Creating Additional Layout



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Additional Layouts in a Report

Introduction

The Report wizard creates the initial layout for your report in the Main Section. When you add queries and groups to the Data Model, you might want to create additional parts of your layout without affecting the layout that is already there. Or you may want to create more than one output of the same data or a subset of the data, for example sending a summary to senior management and a detailed breakdown to individual managers. Both layouts use the same data; there is no need to create two queries.

Report Builder provides the Additional Default Layout tool, which enables you to build up an indefinite number of separate layouts based on all or part of the Data Model.

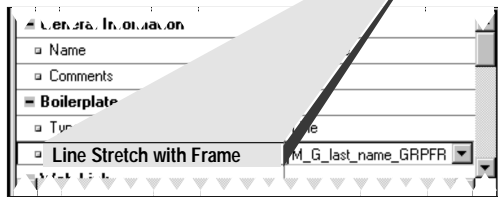
How to Create an Additional Layout

- 1 Choose the section.
- 2 Choose the Additional Default Layout tool from the tool palette.
- 3 Drag an area in the Layout Model editor defining where to place the new objects.
The Report wizard opens.
- 4 Choose the Group tab. Choose the group you want and the direction to print the records.
- 5 Complete the wizard tabs as before.

Flex Lines

Flex lines adjust to fit variable frames and repeating frames

<i>Title</i>	<i>Salary</i>
President	\$2,500.00
VP, Operations	\$1,450.00
VP, Sales	\$1,400.00
VP, Finance	\$1,450.00
VP, Administrative	\$1,550.00



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Variable Length Lines

Introduction

The line drawing tool enables you to draw straight vertical lines between columns of a report. However, these lines are a fixed length, whereas the data in your report might be variable. Report Builder enables you to specify that a line varies in length or width to fill a chosen frame.

Separating Vertical Columns in a Tabular Report

In the report opposite, the job title has a variable number of characters; sometimes the characters require a single line field, sometimes more. The field expands vertically if necessary at run time. The repeating frame also expands to contain the field.

If you draw a vertical line inside the repeating frame, the line is not long enough to span the extra line of characters at run time.

- To make the line stretch to fit the repeating frame, open the property palette. Choose Line Stretch with Frame and select the repeating frame you require, such as the repeating frame, R_XXX.
- To make the line stretch to fit the group frame, draw the line from the top of the column labels through the repeating frame. Change the property Line Stretch with Frame to the group frame, M_XXX_GRPFR.

Live Previewer Versus Layout Editor

You can create and modify a line in the Live Previewer or in the Layout editor. There are advantages and disadvantages to both:

- Live Previewer: It is easy to create a line and change the property. You can easily see the line stretching with the data, but you cannot move the line outside its enclosing object, because you cannot disable Confine Mode.
- Layout editor: You can disable Confine Mode to change the position of the line, but you cannot see the effect of Line Stretch with Frame until you view the report in the Live Previewer.

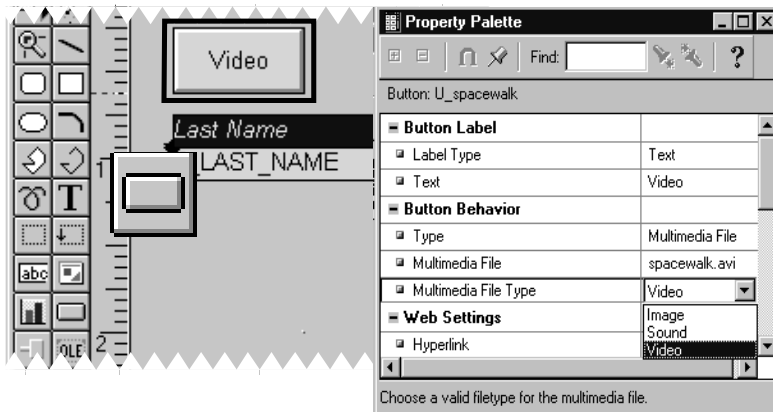
Creating a Button

Buttons

- Provide run-time user interaction for Previewer reports only
- Do not appear in printed output
- Can exist in any layout region
- Perform one of three actions:
 - Display multimedia information
 - Access a URL
 - Drilldown to a detail report

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Displaying a Video Clip from a Button



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Displaying a Multimedia File from a Button

Introduction

Buttons allow users to interact with a report in the Previewer at run time. Buttons have no meaning in a printed report.

Create a button in any report region to:

- Display multimedia information related to report items
- Access a URL
- Drill down to a separate detail report

How to Create a Button

- 1 Choose the Button tool in the tool palette.
- 2 Click and drag an area in the layout.
- 3 Rename the button.
- 4 Choose label type: Text or Icon.
- 5 Enter a text string or an icon name (without the extension).
- 6 Set other properties as required.

Note: Report Builder uses an environment variable to search for icon files outside the current working directory. For Windows this is `UI_ICON`. On UNIX this variable is called `TK_ICON`.

Displaying a Video Clip

Set the following button behavior properties:

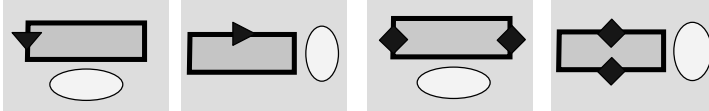
- Type: Multimedia File
- Multimedia File: Name of file
- Multimedia File Type: Video

Testing Button Behavior

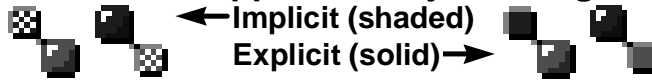
To test that the button works as you want, run the report using the Reports Runtime; buttons are not active in the Live Previewer because this is the *development* previewer.

Creating Explicit Anchors

- **Objects in the push path have implicit anchors (not visible in layout model).**



- **Explicit anchors override implicit anchors.**
- **All anchors appear in Object Navigator.**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating Explicit Anchors

Introduction

The absolute position of each layout object is, by default, related to the position of its enclosing object at run time, unless one of the following is true:

- Report Builder determines that the position causes a conflict with other objects, so Report Builder creates an implicit anchor
- You decide to override the default position of an object, so you create an explicit anchor

Push Path

Report Builder creates an implicit anchor for any object that it considers to be in the push path of another object. For example:

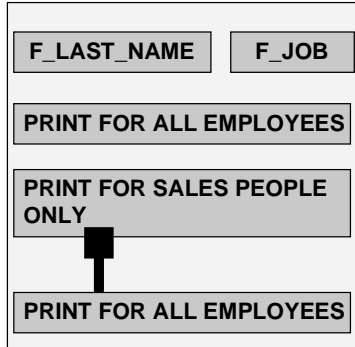
- An object occurring below a vertical repeating frame
- An object occurring to the right of a horizontal repeating frame
- An object occurring below an object that has a variable vertical size
- An object occurring to the right of an object that has a variable horizontal size

Contrasting Implicit and Explicit Anchors

Implicit	Explicit
Not visible in the Layout Model	Visible in the Layout Model
Created and maintained automatically by Report Builder	Created by the developer using the anchor tool in the tool palette
Visible in the Object Navigator if Anchoring Information is enabled (Navigator option)	

Removing Spacing

Layout



Output



- Create anchor from child to parent
- Collapse Vertically = Yes

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Suppressing Space in Conditional Reports

You can hide objects in your report by creating a format trigger to test for a condition and decide whether or not to display the object (Format triggers are covered later in the course.) In some layouts, this can result in unwanted white space between the remaining objects. You can create explicit anchors to remove this extra space.

Removing Spacing in a Conditional Letter

The Problem The layout opposite is a letter that contains three paragraphs of text. Each paragraph is a separate boilerplate object.

The second paragraph is conditional on the employee job title. Conditional formatting is used to hide the field if the title is like 'Sales%'.

Since all three paragraphs are fixed-size objects, their position is directly related to the enclosing, repeating frame. The position of paragraph 3 remains the same, whether paragraph 2 prints or not, which results in a large gap for most letters.

The Solution Create an anchor from the top edge of paragraph 3 (the child object) up to the bottom edge of paragraph 2 (the parent object). Click once on the child to start the anchor, and double-click the parent to complete the anchor.

The start position of the child depends on the final position of the parent. If paragraph 2 does not take up any space, paragraph 3 starts higher up the page.

Collapsing Anchors

The solution suppresses the space that was filled by paragraph 2, but it still shows the gaps *between* both paragraphs. If you want to suppress this space, open the property palette of the anchor and change Collapse Vertically to Yes.

Summary

- **Modifying the Layout Model**
 - Report regions
 - Multipanel reports
 - Layout objects and tools
 - Additional layouts
- **Example reports**
 - Variable length lines
 - Display a video clip from a button
 - Suppress space with an anchor

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Introduction

In this lesson you learned to modify a report layout in the Layout Model editor.

Modifying the Layout Model

- There are two report regions, Margin and Body, and three sections, the Header, Main, and Trailer sections.
- You can create multipanel reports by modifying the logical panel properties of the report.
- You can use the tool palette to create standard drawing objects and report objects. The tool palette also provides some useful editing tools, such as the Frame Select tool.
- You can create many additional layouts in the same report using the Additional Default Layout tool.

Example Reports

This lesson discussed solutions to the following report requirements:

- Create variable length lines
- Display a multimedia file from a button
- Suppress conditional space with an explicit anchor

Practice 10 Overview

- **Adding header and trailer pages**
- **Adding a button to display a multimedia column**
- **Adding a repeating frame to the header page**
- **Adding flexible lines**
- **Anchoring a boilerplate object to an outer frame**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 10

This practice session contains:

- Opening an existing report to add header and trailer pages
- Modifying an existing report to add a button to display a multimedia column
- Modifying an existing report to add a repeating frame to the header section displaying order information
- Opening an existing report and adding flexible lines
- Modifying an existing report to anchor a boilerplate object to an outer frame

Introduction

In this practice session, you use reports created in earlier practice sessions. The aim is to familiarize you with the different layout regions of a report. You also add flexible lines to a frame and anchor one object to another.

Practice Session: Lesson 10

- 1 Open report `p9q4.rdf`. Modify the report as follows:
 - a Create a header section.

Display the `summitlo.tif` file as fixed boilerplate in the body region.
Add a rectangle boilerplate object. Give it fill color.
Place the logo on top of the rectangle. Center the two objects.
Hint: Use the Arrange option from the menu.
 - b Add a report title in the body region.

Use font Arial, size 18 bold.
 - c Add a trailer section that displays today's date with the text "End of report run on: " in the body region. Choose a larger font.
Hint: Copy and paste the date from the Margin Area of the report.
 - d Run and test the report.
 - e Save the report as `p10q1.rdf` and close it.
- 2 Open report `p9q3.rdf` and modify it to display the product picture in a separate window.
 - a Delete field `F_FILENAME` from the body region of the main section in the Layout Model (not the column from the Data Model).
 - b Add a button to display the picture from the multimedia column.

Give it a meaningful label. Change the name to `PICTURE_BUTTON`
 - c Resize the other layout fields to 1/4 inch in height.
Hint: Use the Arrange option from the menu.
 - d Make the button the same size.
Hint: Use the Arrange option from the menu.
 - e Give all fields and the button the same vertical alignment.
Hint: Use the Arrange option from the menu.
 - f Resize the repeating frame and all enclosing frames to be just large enough to fit the fields inside.
Hint: Think of the Flex Mode.
 - g Save the report as `p10q2.rdf` and close it.
 - h Run and test your report. Do not use the Live Previewer; use the Reports Runtime

-
- 3 Open report `p9q1.rdf`.
 - a Modify the report to list all orders and order dates contained in the report in the body region of the header section.
 - b Save the report as `p10q3.rdf` and close it.
 - 4 Open report `p5q3.rdf`.
 - a Use the Report wizard to give `last_name` and `first_name` a width of 6. (You can also do it directly in the Layout Model or from the Live Previewer.)
 - b Add flexible lines vertically between the fields.
 - c Run and test the report.
 - d Save the report as `p10q4.rdf`.

If You Have Time

- 5 Continue with report `p10q4.rdf`.
 - a Add boilerplate and anchor it to half way down the right side of the report group frame.

Hint: To ensure that you get the right frame, select the frame `M_G_LAST_NAME_GRPFR`. You can use either the Object Navigator or the Select Parent Frame tool. Pull the frame out slightly to the right. After you have fastened your anchor, you can make it smaller again.
 - b Run and check that the text is in fact halfway down the data output.
 - c Rerun the report, this time displaying only 10 rows. The boilerplate should still be halfway down the data output.
 - d Save the report as `p10q5.rdf` and close it.

Modifying Properties

Objectives

At the end of this lesson, you should be able to do the following:

- **Create a link file**
- **Modify common layout properties**
- **Modify specific layout properties**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

All Report Builder objects have properties that provide default settings, which in many cases are suitable for your report. However, you can modify these property values; for example, to change the pagination of objects in the report output.

In this lesson you learn how to reference the contents of a file at run time how to and control the placement and pagination of this and other objects in the report output by modifying the object properties.

Objectives

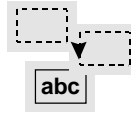
At the end of this lesson, you should be able to do the following:

- Display the contents of a file in the report layout
- Modify layout properties common to all types of object
- Modify layout properties specific to one type of object

Selecting an Object in Lower Layers

Accessing a property palette

- **Object Navigator**
 - **Select object**
 - **Double-click on node icon**
- **Live Previewer or Layout editor**
 - **Use Select Parent Frame tool**
 - **Choose Tools→Property Palette**



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Comparing Properties

Intersection

Freeze

Property Palette		Property Palette	
Object: F_last_name, F_title		Object: F_salary	
* General Information		* General Information	
Field		Field	
Source	Source	Source	salary
Visible	Yes	Visible	Yes
Column		Column	
Datatype	Varchar2	Datatype	Number
Width	25	Width	11,2
Value if Null		Value if Null	
Set Break Order	Yes	Set Break Order	Yes
Break Order	Ascending	Break Order	Ascending
Read from File	No	Break Order	Ascending

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying Layout Object Properties

Introduction

You have already seen that you can access the property palette for any object in several different ways. For layout objects, you can access the same properties by choosing the object in one of the following windows:

- Object Navigator
- Layout Model
- Live Previewer

Selecting an Object in Lower Layers

Because there are several layers in even a simple report, it is sometimes difficult to select the correct object in the Layout Model or Live Previewer, when other objects overlay the object you want.

- Using the Object Navigator:
 - Choose the object directly in the Object Navigator, where you can easily see the name and type of object.
 - To open the property palette, choose the node icon to the left of the object name, or use the right mouse menu.
- Using the Select Parent Frame tool:
 - Choose the top object in the Layout Model or Live Previewer. Choose Select Parent Frame to select the enclosing object on the layer below. For example, select a field; choose Select Parent Frame to select the repeating frame; choose Select Parent Frame again to select the group frame.
 - To open the property palette, choose Tools—>Property Palette or use the Object Navigator.

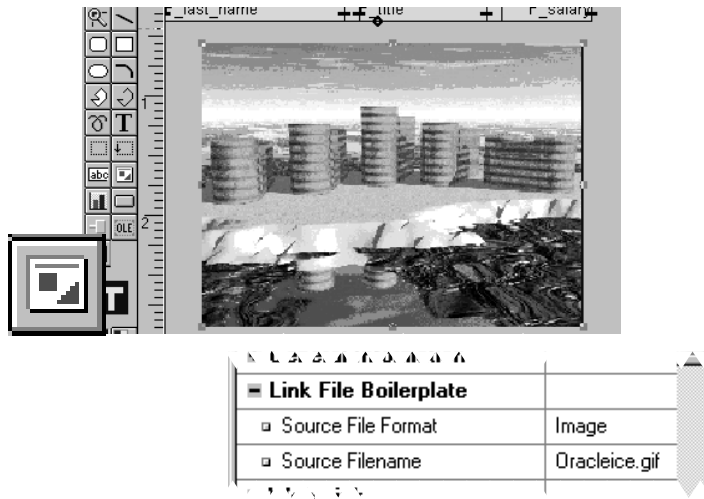
Do not double-click on the object in the Live Previewer, because you will probably select the wrong object on the top layer.

How to Compare Properties of Two Objects

- 1 Select one or multiple objects and open the property palette.

Note: Selecting multiple objects results in an *intersection* of properties.
- 2 Choose the Pin tool to “freeze” the property palette.
- 3 Select the second object and open the property palette.
- 4 Reposition the property palettes to view them side by side.

Displaying File Contents



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Displaying File Contents with a Link File

Introduction

There is a special type of boilerplate object, called a *link file*, that does not have to remain constant for each report run.

A link file allows you to create an external boilerplate object that displays the contents of a file.

Link File Properties

The specific properties for this type of object are as follows:

Property Name	Description
Source File Format	The type of file contents; contents can be Text, Image, CGM, Oracle Drawing Format, or Image URL
Source Filename	The name of the file that you want to link to the report

You can see the full list and descriptions of valid image formats under File—>Import—>Image.

In character-mode reports, Text is the only valid format.

How to Create a Link File Object

- 1 In the Layout Model toolbar, choose the Link File tool. Drag an area in the layout to create a link file object of the required size.
- 2 In the property palette, change the name of the object.
- 3 Enter the name of the link file that you want to include in the output, and the format of the file.

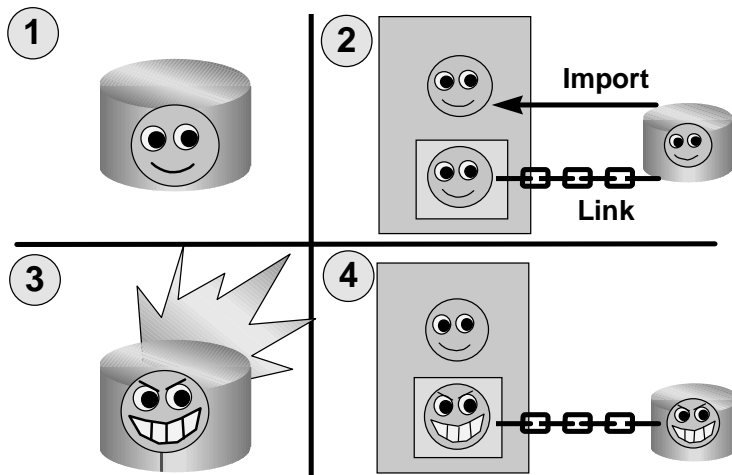
When Are Contents Updated?

Report Builder reads the file contents when you:

- Modify property palette
- Open report definition
- Run report

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Linking and Importing Files



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

When Are the Contents Updated?

Oracle Reports picks up the contents of a link file at run time; therefore the file contents can change dynamically for each report run.

The link file object automatically redisplay the current contents of the file when you:

- Modify the source filename in the property palette
- Open the report (in Report Builder)
- Run the report

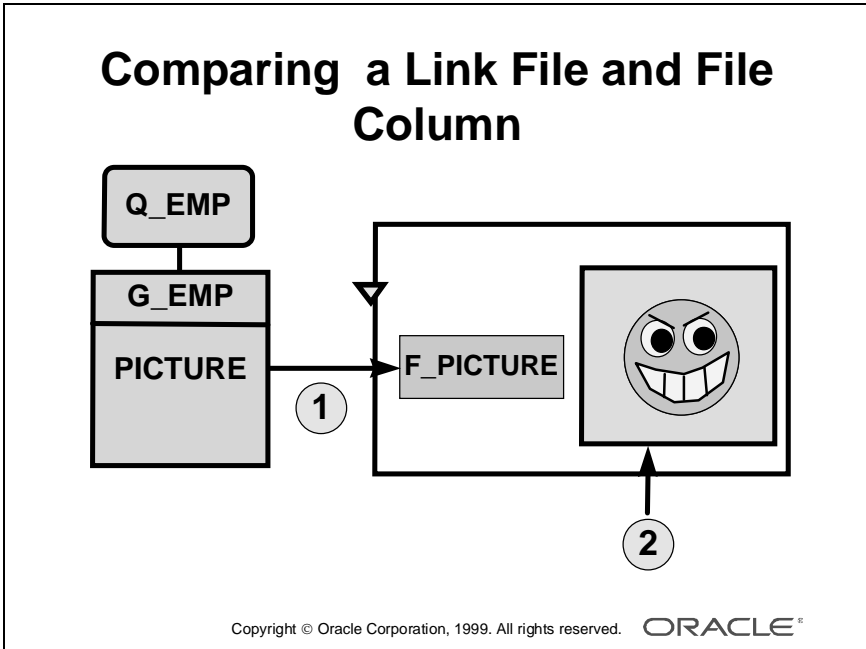
Note: the file must exist at run time; otherwise the report fails with an error message.

Choosing Between Linking and Importing File Contents

If the contents of a file might change, and especially if you want to use the same contents in many reports, use a link file so that the contents are included dynamically at run time.

If the file contents are not likely to change often, you can import the contents.

Note: An imported image is stored in the report definition file, and therefore increases the size of the file. However, the report runs faster if the image is imported rather than linked to a file.



1	The layout field F_Picture depends on its source column PICTURE in the Data Model. The filename, and therefore the picture, can be different for each record.
2	Link File exists as a layout object only. The same picture repeats for each record.

Comparing a Link File and a File Column

The main difference in creating a link file and a file column is that you create a link file in the Layout Model, whereas you create a file column in the Data Model.

The following table shows the differences between a link file and a file column.

A Link File	A File Column
Is an explicit boilerplate object	Is a database column.
Represents one file only	Represents a different value for each record
Displays the same file contents for every occurrence of a repeating frame	Can display different file contents for each record
Exists in the Layout Model	Exists in the Data Model and is the source of a layout field
Is a layout boilerplate object that displays the file contents in the Layout editor during development	Sources a layout field that does not display the file contents in the Layout editor during development; you see the field name only

Layout Object Properties

Four objects with common properties:

- **Frame**
- **Repeating Frame**
- **Field**
- **Boilerplate**

Some common properties affect:

- **Sizing**
- **Pagination**
- **Frequency of display**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying Common Layout Properties

Introduction

Most layout objects share a number of common properties. This section explains how to use these common properties.

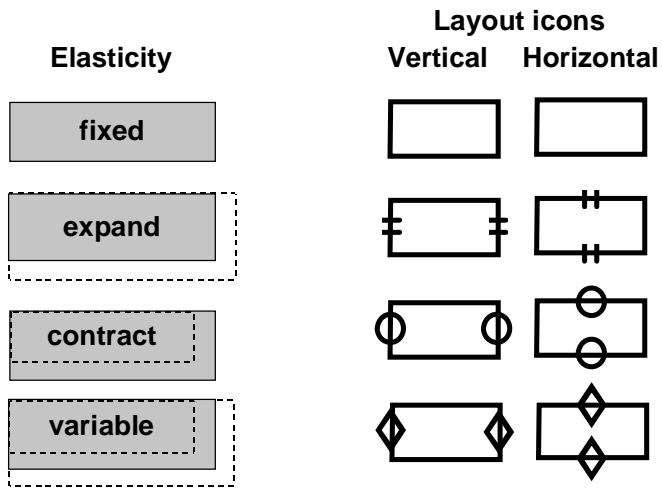
The Four Layout Objects That Have Common Properties

- Frames
- Repeating Frames
- Fields
- Boilerplate Objects

Some Common Properties

- Sizing: Horizontal and Vertical Elasticity
- Pagination: Page Break Before, Page Break After, Page Protect, and Keep with Anchoring Object
- Frequency of Display: Print Object On and Base Printing On

Sizing Objects



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Sizing Objects

Introduction

You can create objects of any size in the Layout Model. The elasticity properties allow you to specify whether the object can change size at run time.

The Four Types of Object Elasticity

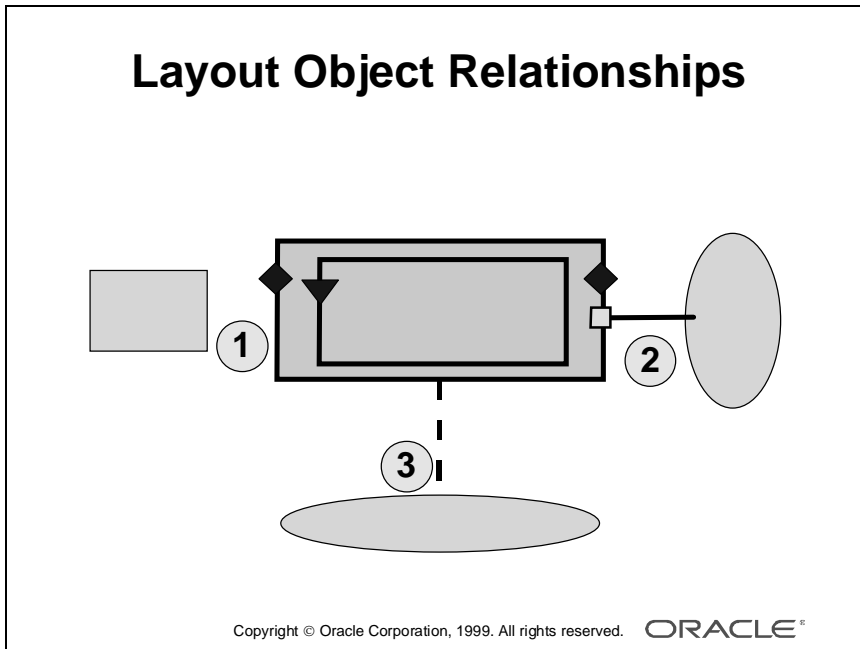
Elasticity	Output Object Size
Fixed	Identical to layout object size
Expand	Can be larger than layout, but not smaller
Contract	Can be smaller than layout, but not larger
Variable	Can be larger or smaller than layout

How Wizard Report Styles Affect Elasticity

Frames: The wizard sets the elasticity of enclosing objects, such as a group frame, to allow for multiple occurrences of a repeating frame.

Fields: When you reduce the size of a character field in the Labels tab, the wizard, depending on the report style, sometimes alters the vertical elasticity of the field to allow for additional characters.

Layout Style	Horizontal Elasticity	Vertical Elasticity
Tabular	Fixed	Expand
Group Above	Fixed	Expand
Form	Fixed	Expand
Form Letter	n/a	n/a
Mailing Label	n/a	n/a



1	No relationship, no anchor
2	Explicit anchor to enforce relationship
3	Implicit anchor because object is in the push path of a frame

Layout Object Relationships

Introduction

To understand how the pagination properties work, you must remember that all objects in the Layout Model are related in a hierarchical structure.

In a previous lesson you learned that the relationship depends on anchors, either explicit or implicit. Each object is anchored either to the logical page or to another object.

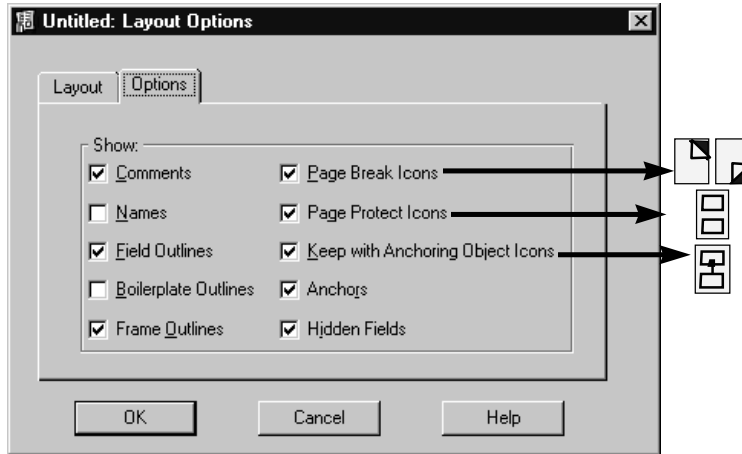
You can use this hierarchical relationship to control the display of related objects, for example if you want to force a page break between related objects, or keep related objects together on a page.

More About the Push Path

By default, Report Builder implicitly anchors each object to its immediate enclosing object. However, sometimes Report Builder must alter the implicit anchor, to ensure that objects do not overlap at run time when their size is not fixed.

Report Builder uses an implicit anchoring algorithm to determine whether an object lies in the push path of another object.

Pagination Icons in the Layout



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

The Four Pagination Properties

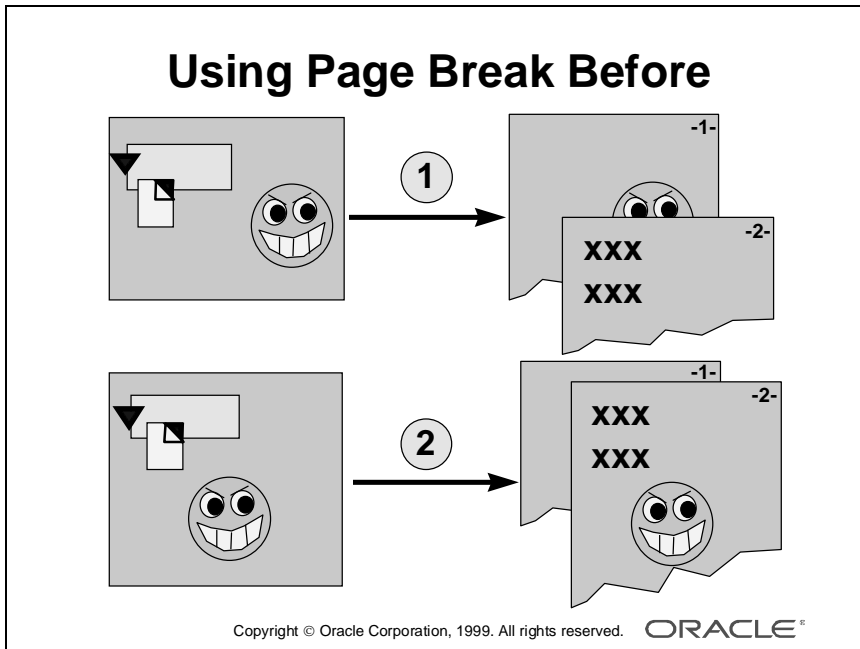
The four common properties that affect whether and where additional page breaks occur in your report are:

- Page Break Before
- Page Break After
- Page Protect
- Keep with Anchoring Object (Advanced Layout)

How to Change Pagination Settings in the Layout Model

The Layout Model displays icons representing each of the properties when they are set to Yes (this is the default). These icons are very useful, especially for maintenance and support. However, you can suppress these and other icons in the Layout Model options.

- 1 Choose Format—>Layout Options—>Layout Model.
- 2 Choose the Options tab.
- 3 Select or clear the options as required.



1	Image is not related to repeating frame
2	Image is in the push path of and implicitly related to the repeating frame

Using Page Break Before

Page Break Before delays the formatting of the current object and its anchored child objects until the next page. Objects that are not implicitly or explicitly anchored to the current object do not move to the next page, even if they appear below the current object in the layout.

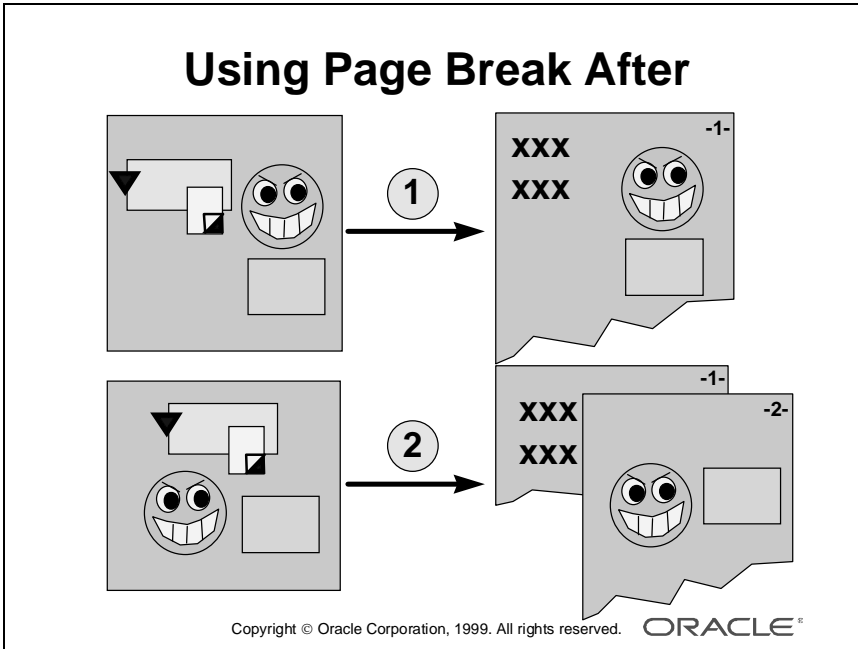
Note: Remember to use the Navigator options to view implicit and explicit anchor information in the Object Navigator. This helps you to understand the relationship between layout objects, and which objects are affected by setting a page break.

Examples

Example 1: The repeating frame and image are not related; therefore, when you select the Page Break Before check box for the repeating frame, the image does not move to the next page.

Example 2: The image is below, and in the push path of, the repeating frame; therefore there is an implicit anchor between the repeating frame (parent) and the image (child). The image formats after the repeating frame on the same page as the repeating frame.

Note: Setting Page Break Before on a repeating frame gives a page break before the first occurrence of the repeating frame only. It does not provide a page break between each record.



1	Image and rectangle are not related to repeating frame; Page Break After has no effect
2	Image and rectangle are in the push path of and are implicitly related to the repeating frame

Using Page Break After

Page Break After delays only the formatting of those objects that are anchored either implicitly or explicitly to the current object.

Use Page Break After when you want to move multiple related objects to a new page. This is quicker and easier to maintain than setting Page Break Before on each of the individual child objects.

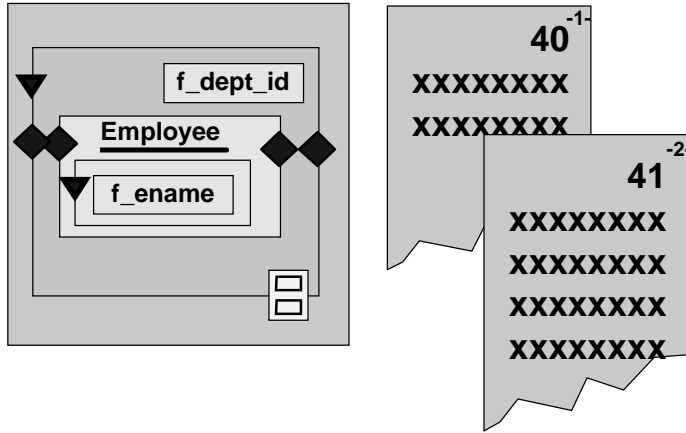
Note: Remember to use the Object Navigator to view implicit and explicit anchor information.

Example Example 1: The repeating frame and image are not related; therefore, when you set Page Break After to Yes for the repeating frame, the image does not move to the next page.

Example 2: The image and rectangle are below, and in the push path of, the repeating frame; therefore there is an implicit anchor between the repeating frame (parent) and the image and rectangle (children). The page break occurs immediately after the last instance of the repeating frame, and the image and rectangle appear on the new page.

Note: Setting Page Break After on a repeating frame gives a page break after the last occurrence of the repeating frame only. It does not provide a page break between each record.

Using Page Protect



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Using Page Protect

Page Protect indicates whether to keep the entire object and its internal child (enclosed) objects on the same logical page.

Page Protect applies only to the first logical page on which the object normally formats; Report Builder ignores Page Protect on subsequent pages to avoid an infinite loop.

Example A particular instance of a master repeating frame might contain so many detail records at run time that it is not possible to display them all on the same page. If so, you can force a page break before the master instance begins.

- To start the master on a new page, set Page Protect to Yes on the master repeating frame.
- If there is no suitable frame to protect several objects, create your own frame, enclose the objects in the frame, and set Page Protect on the frame.

Hint: If you create your own frame to protect several objects, give the frame a solid fill so that you can ensure that you place it at the correct layer of the layout. The frame must be behind all the objects it encloses.

Remember to remove the fill when you are satisfied with the layout.

Ensure that the frame is variable, expand, or contract, depending on the sizing of the objects it encloses.

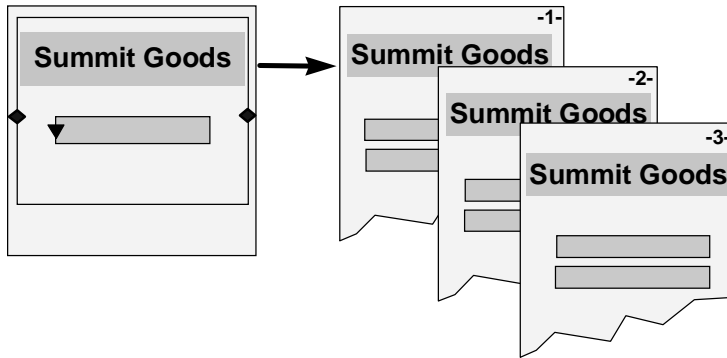
Using Keep with Anchoring Object

The Keep with Anchoring Object property is similar to Page Protect, except that it affects anchored objects instead of enclosing objects.

Keep with Anchoring Object is useful when you want to keep two objects together on the same page.

If there is not enough room to display the child object on the same page as its parent, Keep with Anchoring Object moves both parent and child objects to a new page.

Controlling Print Frequency



- **Print Object On = All Pages**
- **Base Printing On = Enclosing Object**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Controlling Print Frequency

Two related properties, Print Object On and Base Printing On, control how often to print the object in relation to another object.

Which Page?

Print Object On determines on which of the parent’s logical pages the object is printed.

Print Object on	Object prints on
All Pages	All logical pages of the parent
All But First Page All But Last Page	All logical pages except the first or last page
First Page/Last Page	The first or last logical page only
Default	The option that Report Builder chooses; Report Builder resets the property, using an internal algorithm; use the default setting if you have made a mistake and want to return to the report default

Which Parent?

Base Printing On determines whether the parent object is to be the Enclosing Object or the Anchoring Object.

Example

The report opposite shows a text title inside the group frame. The output for this frame shows there are several records that span three pages.

When you create a piece of text, by default it is printed on the First Page only of its enclosing object. In this case the enclosing object is the group frame. You can modify the properties to choose the pages on which the text is printed:

To print the text on all pages of the group frame, set Print Object On to All Pages; set Base Printing On to Enclosing Object (default)

Note: If the text is outside all group frames, its parent is the body page itself; therefore, you can use these two properties to display “Continued...” on all but the last page, or “...continued” on all but the first page.

There are several restrictions that apply to these settings. If you receive an “invalid setting error,” look at the explanation and restrictions in the Report Builder Help Topics.

Properties of a Field

- **Fields define formatting attributes and logic**
- **Wizard creates a field for each column**
- **You create additional fields in Layout editor**
 - **Source: Column, Parameter, or System Variable**
 - **Format Mask: Standard Oracle format symbols**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying Properties of a Field

Introduction

A field is a layout container for a column or a parameter.

A field defines formatting attributes or logic for the data of its related column or parameter.

Fields and Their Relationship to Data Model Objects

The Report wizard creates one field for each column and places these fields inside a repeating frame or at report level.

You can create a field in the Layout editor and relate it to a Data Model column or parameter.

Defining the Source of a Field

The Source property provides a list of columns, of which there are three categories:

- Columns in the Data Model
- Parameters (always listed in uppercase)
- System variables such as today's date and current page number

The datatype of a field depends on its source.

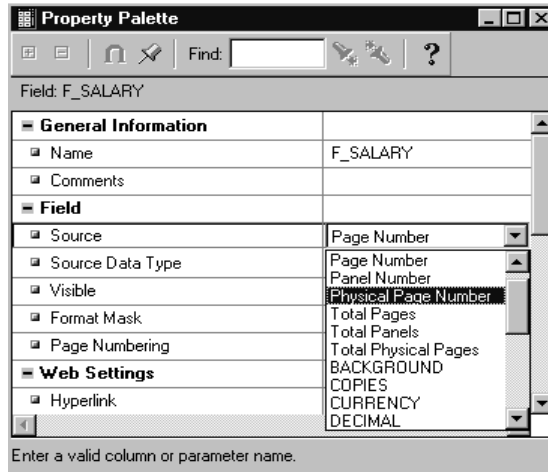
Defining a Format Mask

Remember, when you create a page number object in the Live Previewer, Report Builder provides sample dates to enable you to pick a format easily.

The format mask property appears in the standard Oracle date format symbols, such as MM, which displays the number of the month, or Month, which spells out the name of the month in full.

The list provides all the format masks that exist in your preferences. Modify your preferences to include the common masks that you use, or enter a different mask in this property.

System Variables

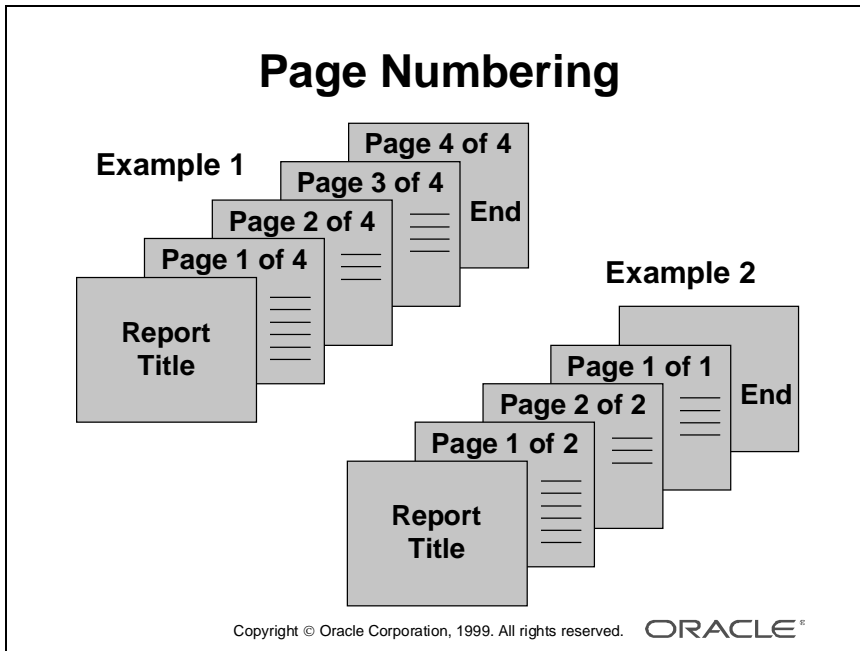


Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

System Variables as the Source of a Field

The following table lists the variables and describes their display values.

Variable Name	Description
Current Date	The operating system date, at run time, after the Runtime Parameter Form has been displayed
Page Number	The current page number, based on the output numbered by logical page
Panel Number	The current panel number in the current logical page
Physical Page Number	The current page number based on the output numbered by physical page
Total Pages	The total number of logical pages in the current report run
Total Panels	The total number of panels in a logical page
Total Physical Pages	The total number of pages based on the output numbered by physical page



Page Numbering

When a field's source is one of the system variables for page or panel numbers, you can define where and how your page numbers are calculated by choosing the Page Numbering property.

Example 1 To number all the pages in the main section of a report and include trailer section pages but exclude header section pages, you would specify:

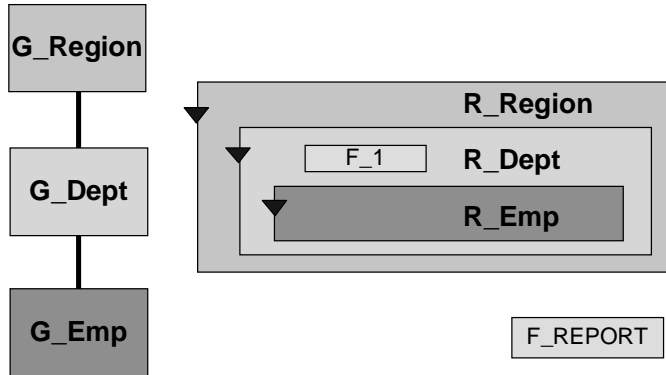
Page Numbering Property	Value
Include	Main Section, Trailer Section
Start At	1
Increment By	1
Reset At	Report

Example 2 To number all pages in the main section of a report, excluding header and trailer pages, and to restart the numbering at 1 for each new department record in the repeating frame R_Dept, you would specify:

Page Numbering Property	Value
Include	Main Section
Start At	1
Increment By	1
Reset At	R_dept

Note: When resetting page numbers for each record in a repeating frame, make sure that each record starts on a new page. Use the Maximum Records per Page property; this property of the repeating frame is discussed later in this lesson.

Valid Source Columns



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Valid Source Columns

When creating new fields, make sure that they exist at the correct level within the report; otherwise frequency errors occur.

If you create a field in a repeating frame, the column you choose as the source must be in the relevant group for the repeating frame or in an ancestor of that group.

In the diagram opposite, the data model shows a three-level hierarchy:

- G_Region is the parent of G_Dept
- G_Dept is the parent G_Emp

The layout shows the three nested repeating frames, R_Region, R_Dept, and R_Emp.

If you create another field in the R_Dept repeating frame, the source column must come from either G_Dept or G_Region; a column in G_Emp would be an invalid source, because values in the G_Emp group occur more often than R_Dept would be printed.

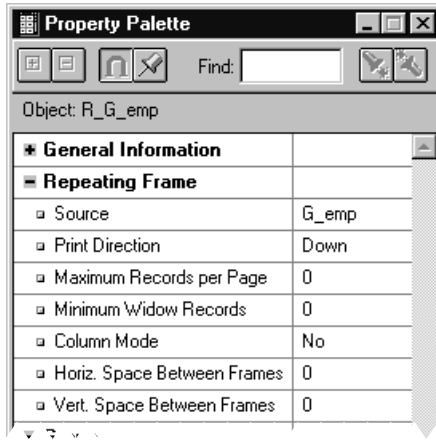
Report-Level Fields

If you create a field outside any repeating frame, that is at report level, the source column must be a report-level column—outside all groups in the Data Model.

System columns, such as current date or page numbers, are report-level columns; you can use these in report-level fields, or at any lower level of your report. For example, you can display the current date at report level or within a repeating frame.

You often use the Header or Margin region to display report-level information.

Properties of a Repeating Frame



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Modifying Properties of a Repeating Frame

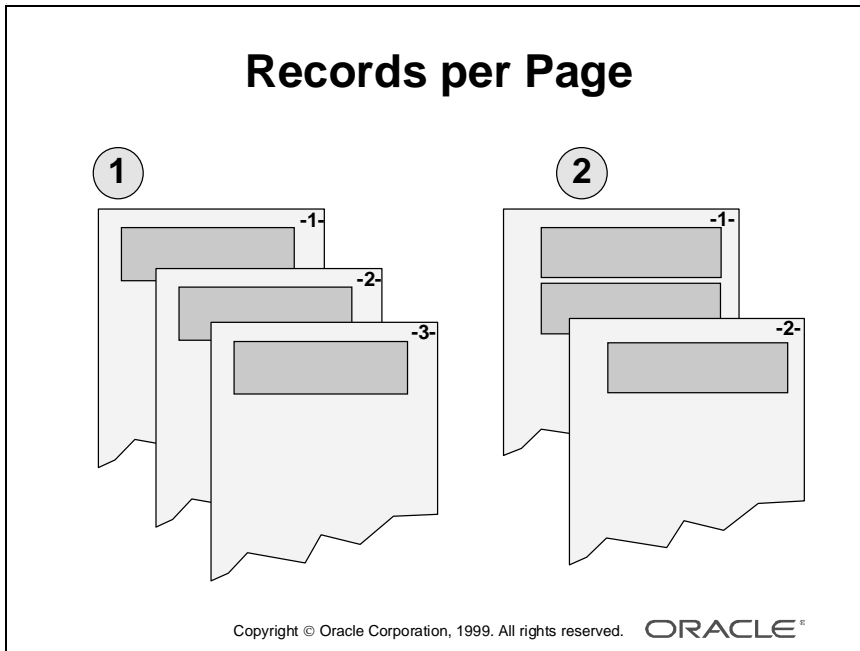
Introduction

A repeating frame is record-related, and its properties allow you to modify specific attributes associated to the display of the records.

Specific Properties of a Repeating Frame

Property Name	Indicates
Source	Name of the Data Model group that provides the source data for this repeating frame; you must source every repeating frame to a valid group
Print Direction	Direction in which records are printed: either Down or Across, Across/Down, or Down/ Across
Maximum Records per Page	Maximum number of records that can be displayed on a single page
Minimum Widow Records	Minimum number of records from a group that can be displayed at the bottom of a page
Column Mode	Whether to maintain the column for each record across multiple pages
Horiz. Space Between Frames Vert. Space Between Frames	Spacing between each record, horizontally and/or vertically

The most commonly used properties are discussed in more detail on the following pages.



1	Maximum Records per Page = 1
2	Maximum Records per Page = 2

Controlling the Number of Records per Page

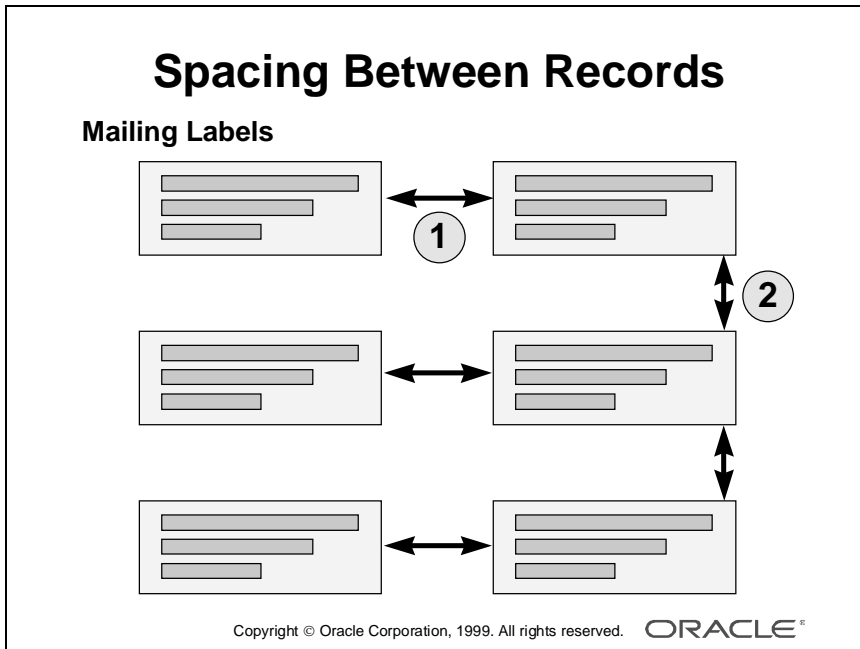
You can specify the maximum number of records that you want to display on each logical page.

In the repeating frame property palette, choose the Maximum Records per Page property. The value you enter for this property can be any positive whole number, or blank.

If this setting is 0 (zero), Report Builder formats as many records as possible on the logical page.

Examples Recall the Form and Form Letter report styles: Report Builder automatically sets Maximum Records per Page to 1. If you require more than one record on each page, you can modify this property.

In the Group Above report style, Report Builder formats as many records as possible on a page (Maximum Records per Page is null). Modify this property for the master repeating frame to display one master record per page.



Controlling Spacing Between Records

The properties Horiz. Space Between Frames and Vert. Space Between Frames control the amount of space between occurrences of a repeating frame; that is, record spacing.

Horizontal Space Between Frames Define the amount of space you want between records horizontally across the page by entering zero or any positive number for this setting.

This setting has no effect if the repeating frame has a print direction of Down.

Vertical Space Between Frames Define the distance you want between records vertically down the page by entering zero or any positive number for this setting.

This setting has no effect if the repeating frame has a print direction of Across.

Note: In bitmapped reports, the distance is a whole or part of the unit of measurement; for example .25 (of an inch).

In character mode reports, the distance is the number of blank characters (horizontal) or lines (vertical). Therefore you must enter a whole number; for example, 10 (characters).

Summary

- **Report objects have many properties**
- **Modify only if necessary**
- **Common properties for most layout objects**
- **Specific properties for some objects**
- **Examples:**
 - **Link file**
 - **Pagination**
 - **Record spacing**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

Report objects provide many properties that you can modify in the property palette to alter the appearance and position of objects in your report output.

You do not have to make many alterations for most standard reporting needs.

- Common properties exist for most layout objects.
- Specific properties exist for individual object types, such as fields and repeating frames.

This lesson explained how to modify object properties and gave examples of some common uses, such as:

- Forcing pagination by setting Page Break Before or Page Break After
- Controlling spacing between records

Practice 11 Overview

- **Modifying the printing of a mailing label report**
- **Controlling the number of records on a page**
- **Adding a link file to a report**
- **Ensuring all details of a master print on the same page**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 11

This practice session contains:

- Modifying the printing of a mailing label report
- Controlling the number of records on a page
- Adding a link file to a report
- Ensuring that all details of a master are printed on the same page

Introduction

Modify a mailing label report to ensure you fill the printed page with label records. You then restrict the number of labels that are printed on each page.

You add a link file to display a company logo by dynamically referencing a file at run time.

Practice Session: Lesson 11

- 1 Open report `p4q4.rdf`.
 - a Modify the layout so that the mailing labels are 2 inches wide and 1 inch high. Use the Arrange option on the menu. Ensure that the company names are printed in alphabetical order across the page.
 - b Give each label a solid fill.
 - c Make the boilerplate object 1.8 inches wide and center it horizontally and vertically in the repeating frame.
 - d Add spacing between the labels: .25 inch between each column of labels across the page; .5 inch between each row of labels down the page.
 - e Save the report as `p11q1.rdf`.
- 2 Continue with report `p11q1.rdf`.
 - a Modify the report so that only six labels are printed on each page.
 - b Save the report as `p11q2.rdf` and close it.
- 3 Open `p9q1.rdf`.
 - a On the operating system, copy the file `summitlo.tif` to a file `logo.tif`.
 - b Delete the logo in the margin area and create a link file in its place to display the contents of `logo.tif`.
 - c Run the report.
 - d On the operating system, copy any other `.tif` file to `logo.tif`.
 - e Rerun your report.
 - f Save the report as `p11q3.rdf`.
- 4 Continue with report `p11q3.rdf`.
 - a Ensure that all items of one order are printed on the same page.
 - b Save the report as `p11q4.rdf` and close it.

Creating and Using Report Parameters

Objectives

At the end of this lesson, you should be able to do the following:

- **Create and reference a parameter**
- **Create a list of values for parameter input**
- **Use and modify a system parameter**
- **Build a parameter form layout**
- **Customize a parameter form layout**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

Parameters allow you to develop dynamic report documents that produce variable report output depending on the parameter value that you enter at run time. This lesson shows how to create and reference user and system parameters and how to customize a parameter form to display parameters at run time.

Objectives

At the end of this lesson, you should be able to do the following:

- Control report output by creating and referencing a parameter
- Control parameter input using a list of values
- Manipulate reports at run time using system parameters
- Provide for user input of parameters by building a parameter form
- Customize a parameter form layout

User Parameters

Can be used to:

- Restrict values in a WHERE clause

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
WHERE ID = <a value>
```

- Substitute any part of select statement

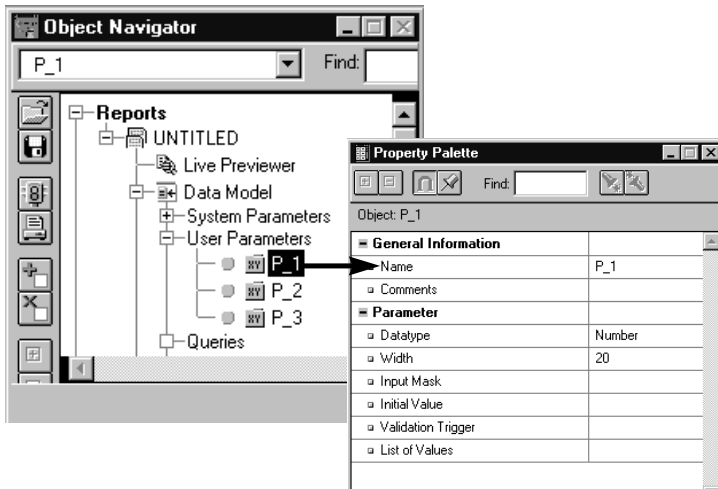
```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
<a where clause>
```

- Substitute a single column or expression

```
SELECT <a column/expression>
FROM S_CUSTOMER
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating a User Parameter



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating User Parameters

Introduction

You can create your own parameters and use them to change the SELECT statement of your query at run time.

What Is a User Parameter?

A *user parameter* is a Data Model object that you create to hold a value that users can change at run time.

You can reference a parameter anywhere in a query. For example:

- Restrict values in the WHERE clause of the SELECT statement
- Substitute any part of the SELECT statement, including the entire statement
- Substitute a single column or expression in the SELECT list

You can display the contents of a parameter in your report by creating a Layout field and entering the parameter name in the Field Source property.

How to Create a User Parameter in the Object Navigator

- 1 In the Object Navigator choose the User Parameter node, then choose the Create tool.
Note: If this is the first parameter, you can create it by double-clicking the User Parameter node.
- 2 Rename the parameter and open the property palette.
- 3 Verify the datatype and width. Enter an initial value if required.

Parameter Properties

Property	Description
Datatype	Specify whether parameter value is character, number, or date; default is Number when created in the Object Navigator.
Width	Specify maximum allowable width of parameter value; maximum is 64KB.
Input Mask	Allow users to enter a numeric or date value using a specific format.
Initial Value	Specify the value to use. You can override this value at run time from the command line or parameter form.
Validation Trigger	Validate the parameter value. Enter a PL/SQL function.
List of Values	Create a list of values from which users select a valid value at run time

Referencing Parameters in a Report Query

Bind reference replaces a value

:parameter_name

Parameter object may be created by default

Lexical reference replaces a clause

¶meter_name

Parameter object may be created by default

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Referencing Parameters in a Report Query

Introduction

There are two ways to reference parameters in a query:

- Use a bind reference
- Use a lexical reference

What Is a Bind Reference?

A *bind reference* replaces a single value or expression.

To create a bind reference in a query, prefix the parameter name with a colon (:).

If the parameter object does not exist, Report Builder automatically creates it for you and displays a message. In this case, the parameter default datatype is CHARACTER, not NUMBER.

What Is a Lexical Reference?

A *lexical reference* is a text string and can replace any part of a SELECT statement, such as column names, the FROM clause, the WHERE clause, or the ORDER BY clause.

To create a lexical reference in a query, prefix the parameter name with an ampersand (&).

If the parameter object does not exist, Report Builder automatically creates it for you and displays a message. In this case, the parameter default datatype is CHARACTER, not NUMBER.

Comparing Bind and Lexical Reference Types

Type	Prefix	Use to Replace	Parameter Created by Default?
Bind	:	Single value or expression in the following clauses: WHERE, GROUP BY, ORDER BY, HAVING, CON- NECT BY, START WITH	Yes, if it does not already exist. Report Builder displays a warning message and adds the parameter to User Parameters in the Object Navigator.
Lexical	&	Any part of a SELECT statement	Yes, if it does not already exist. Report Builder displays a warning message and adds the parameter to User Parameters in the Object Navigator. Datatype must always be Character.

Using Bind References

- Restrict values in a WHERE clause

```
SELECT NAME, SALES_REP_ID  
FROM S_CUSTOMER  
WHERE ID > :P_CUST
```

- Substitute a single value or expression in the select statement

```
SELECT NAME, SALES_REP_ID  
FROM S_CUSTOMER  
ORDER BY DECODE(:SORT, 1, NAME,  
                2, STATE, COUNTRY)
```

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Using Bind References

Use a bind reference anywhere in a query where you can use a single literal value, such as a character string, number, or date.

Example

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
WHERE ID = :P_CUST
```

This statement allows you to enter a specific customer number at run time. For example, if you enter 102, the **WHERE** clause uses the value 102 to restrict data fetched and to fetch the one customer that has ID 102.

Example

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
ORDER BY DECODE(:SORT, 1, NAME, 2, STATE, COUNTRY)
```

This statement allows you to output different versions of the report:

- Enter 1 for the parameter **SORT** to display the output ordered by customer name.
- Enter 2 to display the output ordered by sales representative ID.

Where Can You Not Use Bind References?

You cannot use a bind reference to:

- Replace a column name in the **SELECT** clause, although you can reference a value, such as the contents of a parameter:

```
SELECT LAST_NAME, SALARY * :P_RATE
FROM S_EMP
```

- Replace any part of a **FROM** clause
- Replace reserved words or clauses

Using Lexical References

- Use to substitute any part of the query

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
&P_WHERE_CLAUSE
&P_ORD_CLAUSE

SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
&P_WHERE_ORD_CLAUSE

SELECT &P_CUST CUST, &P_SALESREP REP
FROM &P_TABLE
```

- Ensure that the number of values and datatypes match at run time

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Using Lexical References

Use a lexical reference to replace any clause in a SELECT statement, or even to replace the entire statement.

Examples

- The following statements use lexical references to substitute parts of the query at run time.
- To specify a WHERE clause, ORDER BY clause, or both at run time (as two separate parameters):

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
&P_WHERE_CLAUSE
&P_ORD_CLAUSE
```

- To specify a WHERE clause, ORDER BY clause, or both at run time (as one parameter):

```
SELECT NAME, SALES_REP_ID
FROM S_CUSTOMER
&P_WHERE_ORD_CLAUSE
```

- To specify two column names and the table names at run time:

```
SELECT &P_CUSTNAME CUST, &P_SALESREP REP
FROM &P_TABLE
```

Note: When you use lexical references in the SELECT list, you must, at run time, specify the same number of items of the correct datatype, as defined in the Data Model.

Hints and Tips When Referencing Parameters

Always do the following:

- **Specify column aliases when substituting column names**
- **Create lexical parameters explicitly in the Object Navigator (Report Builder creates bind parameters if necessary)**
- **Enter an initial value for parameters that affect query validation when NULL**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Hints and Tips for Referencing Parameters

Specifying Column Aliases

If you substitute the name of a column or expression in the `SELECT` list with a lexical reference in the `SELECT` list, always add a column alias after the reference. Otherwise, when the parameter value changes at run time, the column name in the `SELECT` list does not match the report column object and the report does not run.

Referencing Nonexistent and Null Parameters

Remember that Report Builder automatically creates a parameter when you make a bind reference to a nonexistent parameter.

When you create a lexical parameter, you might need to include an initial value before referencing the parameter in the query, because depending on your use of the parameter, a `NULL` value can cause a syntax failure.

Example

The following statement does not cause a syntax error when `P_ORDER_CLAUSE` is `NULL`.

```
SELECT . . . FROM . . .  
      &P_ORDER_CLAUSE
```

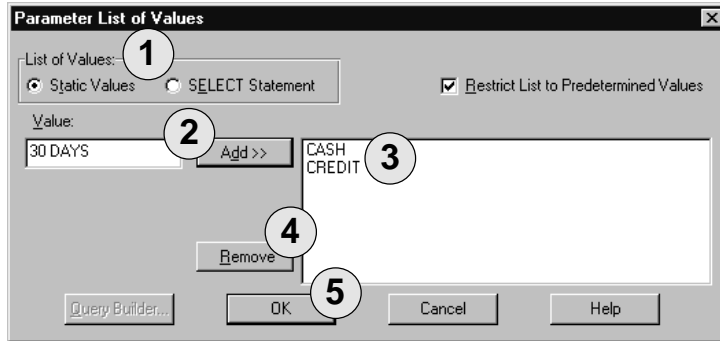
The following statement does cause a syntax error when `P_ORDER_CLAUSE` is `NULL`.

```
SELECT . . .  
FROM . . .  
ORDER BY &P_ORDER_CLAUSE
```

Remember to define an initial value for lexical parameters that affect the validity of the statement, and to validate those that you allow a user to enter at run time.

Note: Always use column aliases when substituting column names with lexical references.

Static List of Values



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Choose Static Values.	4	To remove a value; select a value and choose Remove.
2	Enter a value and choose Add.	5	Choose OK to accept the list and exit.
3	Repeat for each value to build the list of values.		

Creating a List of Values

Introduction

You can create a list of values from which users select a valid value at run time. You can restrict users to only those values in the list, or allow them to enter a different value.

For bind parameters, the list can be a static list of values or a dynamic list that selects values from the database at run time.

For lexical parameters, you can enter a static list of values.

How to Create a Static List of Values

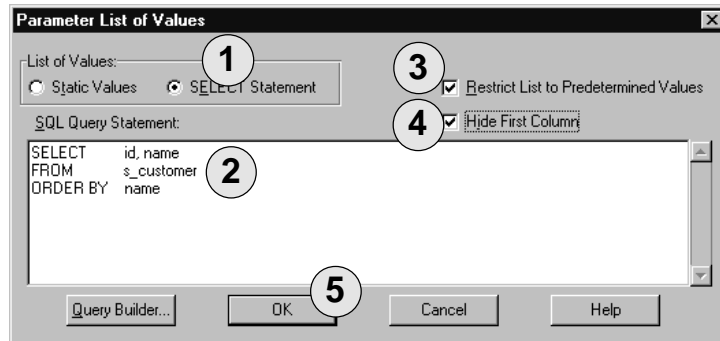
- 1 In the Parameter property palette, choose the List of Values property. The Static Values option button is selected by default.
- 2 Enter a value in the value field and choose Add.
- 3 Repeat for each value you want in the list.

Note: To remove a value, select the value in the list and choose Remove.

Restricting Users to Values in the List

Set the Restrict List to Predetermined Values property to determine whether to prevent users from entering any value not included in your list. By default, this property is Yes. If you set the property to No, users can still select a value from the list, but they can also enter other values or leave the value blank.

Dynamic List of Values



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

1	Choose SELECT Statement.	4	Hide the first column from the list.
2	Enter a valid query.	5	Choose OK to accept the query and exit.
3	Select or clear the Restrict List check box.		

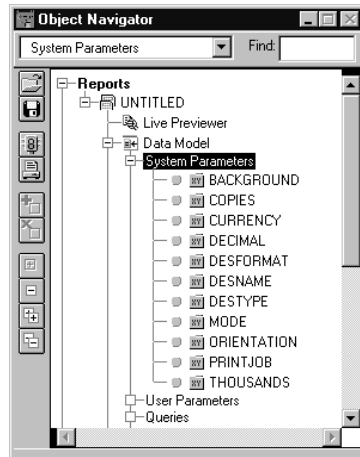
How to Create a Dynamic List of Values

- 1** In the Parameter property palette, choose the List of Values property.
- 2** Choose SELECT Statement.
The SQL Query Statement area is displayed.
- 3** Enter a query to populate the list of values. You can include more than one column. The parameter takes its value from the first column in the list and the column values appear concatenated in the list at run time.
- 4** Set the Restrict List to Predetermined Values property, as required.

Displaying Meaningful Values Instead of Codes

To see a more meaningful list of descriptive values, instead of the primary or foreign key column that you reference in the query, set Hide First Column to Yes. Make sure that the primary or foreign key column is first in the SQL query statement, because this is still the value that the parameter object contains at run time.

System Parameters



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Referencing System Parameters

Introduction

Report Builder provides *system parameters* to allow you to modify standard run-time settings for each report definition.

There are currently eleven system parameters. You cannot delete them.

Each parameter has a default value that you can modify.

Parameter	Use to specify	Values	Default
BACK-GROUND	Whether report is to run in background (in Microsoft Windows, setting this parameter to Yes invokes the Background Engine RWRBE60 if not already running)	Yes, No	No
COPIES	Number of copies if destype=printer	Any integer	1
CURRENCY	Symbol for currency indicator	Max 4 characters	
DECIMAL	Symbol for decimal indicator	Any single char	
DESFORMAT	Output device definition, ignored if type=screen/preview	Examples: dflt, pslan80, pdf,HTML	dflt
DESNMAME	Destination name, such as filename, printer name, mail userid; ignored if type=screen/preview		<report>.lis
DESTYPE	Destination type for output	File, printer, mail, screen, preview	Screen
MODE	Whether report executes in bitmapped or character mode	Bitmap, character	Default
ORIENTATION	The print direction of printer output	Portrait, landscape	portrait
PRINTJOB	Whether print job dialog box is displayed at run time, if destype=file or printer	Yes, No	Yes
THOUSANDS	Symbol for thousands indicator	Any single character	

Displaying a Currency Symbol for Monetary Values

Build:

report:Parameter - CURRENCY

Initial Value £

report:Layout Field - F_Money

Format Mask \$NNN0

£1000
£6000
£5000

FF1000
FF6000
FF5000

Run-time:

RWRUN60 report=fred.rep...currency=FF

Displaying a Currency Symbol for Monetary Values

A common requirement is to format currency values preceded by a currency symbol. You may require the symbol to be £ for pounds sterling, \$ for U.S. dollars, FF for french francs, and so on. You can set different initial values in different reports.

How to Display a Different Currency Symbol

- 1 Enter the currency symbol (\$) in the format mask of one or more layout fields.
- 2 From the Object Navigator, open the property palette for the Currency system parameter and enter a maximum of four characters in the initial value.

Note: Currency, thousands, and decimal symbols appear in the report output only if you use them in the format mask of a layout field.

Altering the Parameter Value at Run Time

You can modify the parameter value at run time in one of two ways:

- Include the currency parameter on the run-time command line
- Enable users to alter the parameter value on the run-time parameter form

If the currency symbol you choose is several characters in length, remember to make the relevant fields wide enough to include it.

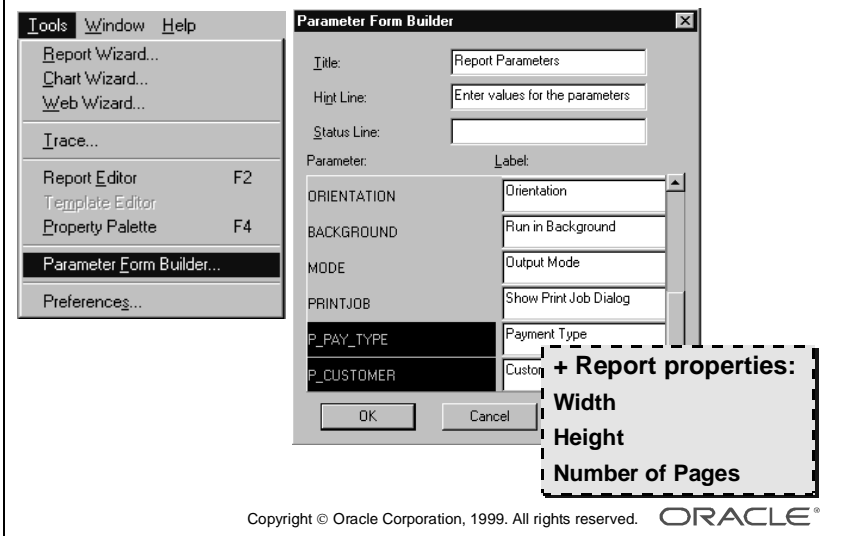
Priority of Parameter Values

RWRUN60: Runtime Parameter Form overrides Command Line. Command Line overrides Initial Value.

RWDES60: Runtime Parameter Form overrides Initial Value. Initial Value overrides Command Line.

Note: When running a report in Report Builder, an initial value for the current report takes priority over a default that you specify in the R30DES command line. Use Tools—>Preferences—>Runtime Values and Runtime Settings for testing.

Building a Parameter Form



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Building a Parameter Form

Introduction

The reports that you created in previous lessons ran successfully without a parameter form. However, when you created parameters, a default run-time parameter form appeared, to enable you to change the values at run time.

You can build your own parameter form, select the parameters that you want to display, and then customize the appearance of the parameter form in the Report editor.

How to Build a Parameter Form

- 1 Select menu item Tools—>Parameter Form Builder.
The Parameter Form Builder dialog box appears.
- 2 If desired, modify the title, hint, and status line text. These lines are displayed at the top of the parameter form at run time.
- 3 Select or deselect parameters in the parameter section as necessary.
Note: Selected parameters are displayed in the dialog box as highlighted on a black background. Deselected parameters do not appear in the parameter form.
- 4 Modify label text of the selected parameters, if you wish.
- 5 Choose OK to create the parameter form.

If you create many parameters in your report, the run-time parameter form can spread over more than one page. You can view subsequent pages at run time by choosing Next.

Setting Report Properties for the Parameter Form

There are three properties in the Report property palette, under the Parameter Form Window node, that specifically apply to the parameter form.

Report Property	Use to
Form Width	Define the width of the parameter form that the user sees at run time.
Form Height	Define the height of the parameter form that the user sees at run time.
Number of Pages	Define the number of physical pages that make up the parameter form.

Customizing a Parameter Form

The screenshot displays the Oracle Developer interface for customizing a parameter form. The main window, titled 'Report Parameters', contains the text 'Enter values for the parameters' and a background image of stacks of coins. Below this, four parameter fields are visible: 'Destination Name' (PF_DESNAME), 'Currency Symbol' (PF_CURRENCY), 'Payment Type' (PF_P_PAY_TYPE), and 'Customer' (PF_P_CUSTOM). A 'Property Palette' window is open on the right, showing the properties for the selected object 'PF_P_PAY_TYPE'. The palette is organized into sections: 'General Information' (Name: PF_P_PAY_TYPE, Comments: empty), 'Parameter Form Field' (Source: P_PAY_TYPE), and 'Parameter' (Datatype: Character, Width: 10, Initial Value: empty, Validation Trigger: empty, List of Values: empty).

Property Palette	
Object: PF_P_PAY_TYPE	
General Information	
Name	PF_P_PAY_TYPE
Comments	
Parameter Form Field	
Source	P_PAY_TYPE
Parameter	
Datatype	Character
Width	10
Initial Value	
Validation Trigger	
List of Values	

Copyright © Oracle Corporation, 1999. All rights reserved. **ORACLE**

Customizing the Parameter Form Layout

You can customize the parameter form layout in a similar way to customizing the report layout. You can modify colors and fill patterns (except in fields), move or delete existing objects, create additional objects, import file contents such as a company logo, and so on.

Creating Parameter Fields

A *parameter field* is a placeholder for a parameter value on the run-time parameter form, in the same way that a field is a placeholder for a column value in the layout.

You cannot modify the color or fill patterns of a parameter field. However, the property palette allows you to modify all parameter properties.

Report Builder creates one field for each parameter that you select in the Parameter Form Builder. If a parameter in your Data Model does not have an associated field, it does not appear in the run-time parameter form because it does not have any display attributes.

You can create additional fields by using the field tool in the parameter form toolbar and sourcing the field to an existing parameter.

Creating Parameter Boilerplate Objects

Parameter boilerplate is text or graphics that either you create or the Parameter Form Builder creates by default.

The Parameter Form Builder creates parameter boilerplate objects for the labels of each parameter field as well as for the title, hint, and status lines.

You can create additional boilerplate objects to customize your parameter form; for example, create report heading pages, modify default labeling, import images, and so on.

Note: You cannot create a link file in the parameter form; however, you can import file contents, as you can in the layout.

Summary

- **Parameter types**
 - **User parameters: bind/lexical, LOVs**
 - **System parameters**
- **Modifying values at run-time**
 - **Command line arguments**
 - **Run-time parameter form**
- **Creating a run-time parameter form**
 - **Build: Parameter Form Builder**
 - **Customize: Report Editor**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

- Parameter types:
 - User Parameters:
References: bind or lexical
List of values: dynamic (bind only) or static
 - System Parameters
- Modifying parameter values at run time:
 - Pass the parameter on the R30RUN command line
 - Allow users to change values in the run-time parameter form
- Creating a run-time parameter form:
 - Build a parameter form with the Parameter Form Builder
 - Customize the parameter form layout in the Report editor

Practice 12 Overview

- **Modifying an existing report to use a bind parameter**
- **Adding a dynamic list of values**
- **Adding a lexical parameter**
- **Adding a static list of values**
- **Creating a customized parameter form**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 12

This practice session contains:

- Modifying an existing report to use a bind parameter
- Adding a dynamic list of values
- Adding a lexical parameter
- Adding a static list of values
- Creating a customized parameter form

Introduction

In this practice session, you create bind and lexical parameters to restrict records in the report. You add lists of values to select valid values at run time.

You then create your own default parameter form and customize it to include extra text and graphics.

Practice Session: Lesson 12

- 1 Open report `p11q3.rdf`.
 - a In the Report wizard add the following columns from the `S_CUSTOMER` table. Create a new group for the customer information above the order group.

Column	Label	Width
name		30
address		30
city		30

- b Create a reference to a bind parameter `P_CUSTOMER` in the query to restrict orders to the parameter value you input at run time. Note how Reports creates the parameter automatically.
 - c Run and test the report with valid customer IDs.
 - d Save the report as `p12q1.rdf`. Do not close it.
- 2 Continue with report `p12q1.rdf`.
 - a Add a list of values to select any valid customer ID in the `S_CUSTOMER` table. Do not allow users to enter any other value. The list should display customer names, not ID numbers.
 - b Run and test the report with different values.
 - c Save the report as `p12q2.rdf`. Do not close it for the moment.
- 3 Open report `p5q3.rdf`.
 - a Create a lexical parameter to enable users to order the data by different columns (or not at all) at run time. Replace the `ORDER BY` clause in the query with a lexical reference to the parameter. Type in an initial value for the parameter to order by the employee's last name.
 - b Add a static list of values. Code two or three alternative `ORDER BY` clauses. Allow any clause to be entered at run time.
 - c Run the report to test. Select an `ORDER BY` clause from the list. Enter your own `ORDER BY` clause. Delete the default parameter value.
 - d Save the report as `p12q3.rdf` and close it.

- 4** Continue with report `p12q2.rdf`.
 - a** Create a default parameter form.
Include a Title message, but not a Hint message.
Display the parameters `DESNM` and `P_CUSTOMER`.
 - b** Use your imagination to customize the form in the Parameter Form editor.
Ensure that the parameter `P_CUSTOMER` is placed in a prominent position for user entry.
Add some graphics; modify colors and fill patterns.
 - c** Run the report to test.
 - d** Save the report as `p12q4.rdf` and close it.

Embedding a Chart in a Report

Objectives

At the end of this lesson, you should be able to do the following:

- **Create and display a simple chart**
- **Display an existing chart**
- **Modify chart data dynamically**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Overview

Introduction

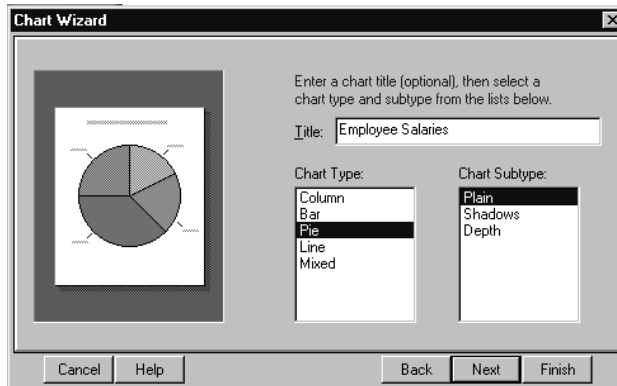
In this lesson you learn to display an Oracle Developer graphical chart in a report at run time.

Objectives

At the end of this lesson, you should be able to do the following:

- Create and display a simple Graphics chart in a report using the Chart Wizard
- Display an existing Graphics chart in a report
- Explain how to modify data dynamically in an existing chart

Creating a Chart



- Image shows chart type
- Subtypes depend on type

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Creating and Embedding a Graphics Chart in a Report

Introduction

Developer/2000 Release 2 provides an easy-to-use Chart Wizard in Report Builder (and Form Builder) to allow you to easily produce a chart that appears in your report at run time.

Invoking the Chart Wizard

In Report Builder, invoke the Chart Wizard in one of the following ways:

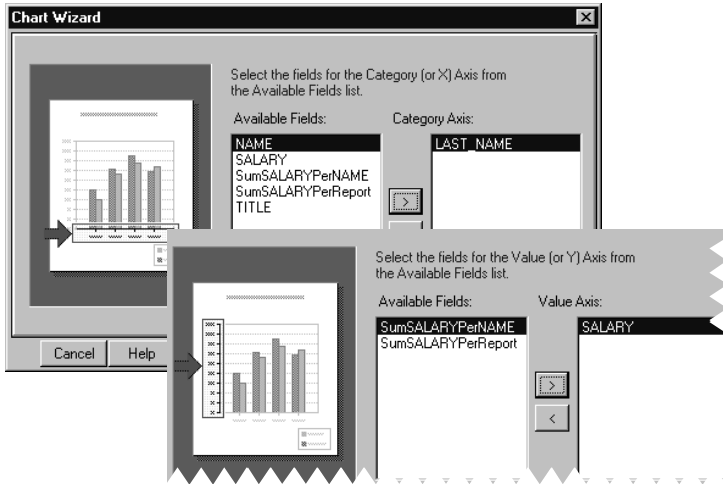
- Choose the Chart Wizard tool in the Live Previewer
- Choose the Chart Wizard tool in the Layout Model
- Choose Tools—>Chart Wizard from the menu

Choosing the Chart Style

Choose the style of chart that you require. The picture on the left of the wizard displays the relevant chart style.

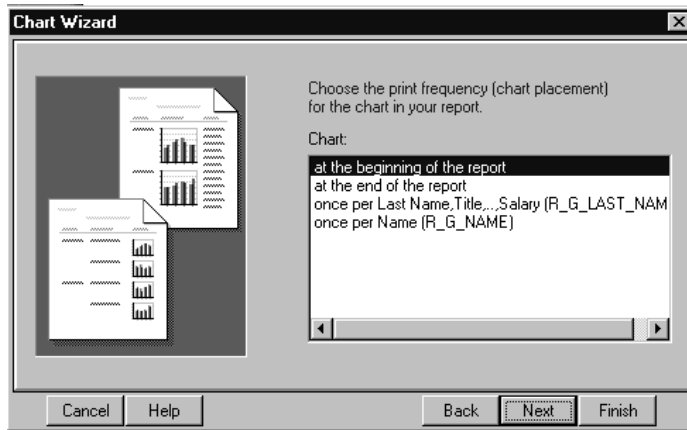
Choose the chart subtype that you require, and then choose Next to continue to the next wizard page.

Choosing the Chart Data



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Positioning the Chart



Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Choosing the Chart Data

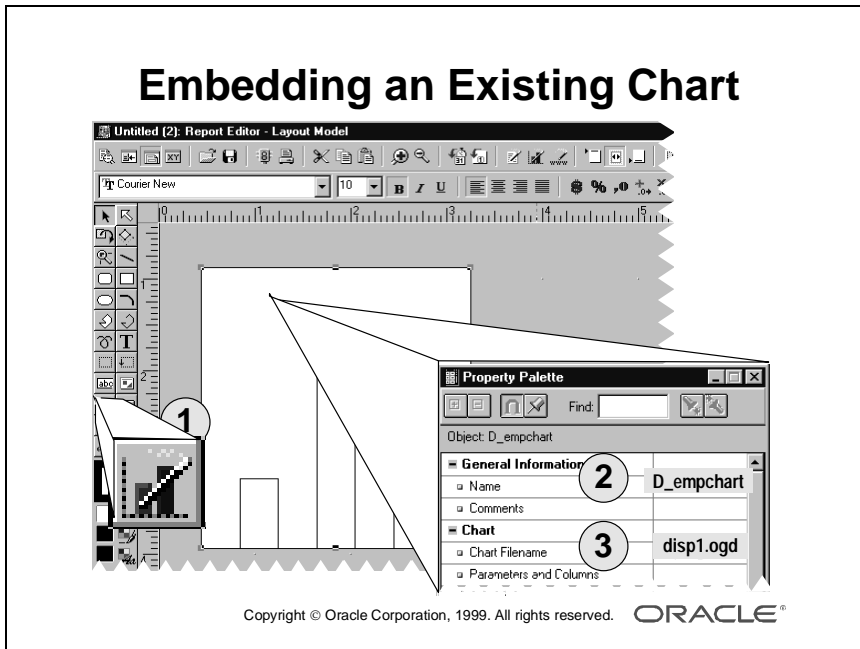
The next two wizard pages allow you to define the data in the chart:

- 1** Select the field that you want to appear as the categories of your chart:
 - Column, Line, Mixed chart: Categories appear along the x-axis
 - Bar chart: Categories appear down the y-axis
 - Pie chart: Each category represents one segment of the pie
- 2** Select the field that you want to appear as the values of your chart:
 - Column, Line, Mixed chart: Values are the height of the columns against the y-axis
 - Bar chart: Values are the length of the columns against the x-axis
 - Pie chart: Values are the size of each segment of the pie

Positioning the Chart in the Report Output

The final page of the Chart wizard allows you to position the chart at the beginning or end of the report output, or to display the chart once for each occurrence of a group.

Embedding an Existing Chart



1	Create the display object
2	Rename the display object
3	Enter the chart filename (created in Graphics Builder).

Embedding an Existing Chart in a Report

Introduction

If you already have a chart that you created or modified in Developer/2000 Graphics Builder and that you want to embed in your report definition, you can create a display object in the report layout and associate it with your chart (.ogd file).

How to Create a Display Object

- 1 In the Layout Model, choose the Chart tool in the vertical toolbar.
- 2 Click and drag an area in the layout where you want the chart to appear.
Note: If you want to repeat the chart for each occurrence of a group, such as for each department, you must expand the repeating frame so that it encloses the display object.
- 3 Rename the display object in the Object Navigator or in the display object property palette.
- 4 In the property palette, enter the chart filename or choose the file from the browser using the button on the right side of the property palette.
Note: If you use the browser, remove the directory path, so that only the filename remains. Make sure that the REPORTS30_PATH variable includes the directory that contains your chart file.

Passing Parameters or Data Records to an Existing Chart

In an existing chart (built using the Graphics Builder), you might have parameters that modify the chart depending on the value entered at run time.

You can pass a parameter from your report to the chart at run time, so that your chart parameter value changes each time Graphics Builder executes the chart.

Alternatively, you can pass entire records from your report to the chart, so that the chart does not query the data from the database.

To map report objects to graphics parameters or columns, in the chart object property palette, choose Parameters and Columns under the Chart node. You must know the names of the graphics parameters and columns that exist in the graphics .ogd file.

For further details, see the online Help.

Summary

- **Build new chart:**
 - Invoke Chart wizard to build and embed a chart
 - Enhance chart in Graphics Builder
- **Embed existing chart:**
 - Create display object
 - Link to chart filename
 - Pass parameters or column values

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Summary

- Use the Chart Wizard to create and embed a simple chart in your report.
- Reenter the Chart Wizard to make changes to the chart object, such as its position in the report.
- Use Developer/2000 Graphics Builder to enhance and modify the chart definition.
- Create a display object in the Layout Model to embed a chart that already exists in an .ogd file. Enter the chart (.ogd) filename.
- Pass parameters or column values from a report to a chart to restrict values or data records if the chart contains a separate SQL query.

Practice 13 Overview

- **Using the Chart wizard to create a chart**
- **Altering the chart in the Chart wizard**
- **Creating charts that print once for each record of a group**

Copyright © Oracle Corporation, 1999. All rights reserved. ORACLE®

Practice Session Overview: Lesson 13

This practice session contains:

- Using the Chart Wizard to create a chart in an existing report
- Altering the chart in the reenterable Chart Wizard
- Creating charts in an existing report that will be printed once for each record of a group

Introduction

In this practice session you open existing reports to add charts using the Chart Wizard. You also reenter the Chart Wizard to change the characteristics for a chart.

Practice Session: Lesson 13

- 1 Open report `p12q1.rdf`
 - a Remove all reference to the parameter `P_CUSTOMER` from the query.
 - b Rename `SumLine_totalPerNAME` to `Total` (a shorter name looks better in a chart).
 - c Use the Chart wizard to create a line chart.

Chart Type	Line
Chart Subtype	Plain
Category	NAME
Value	TOTAL
Position	At the beginning of the report
Filename	<code>p13q1.ogd</code>

- d Run the report to test. Resize the chart if necessary.
 - e Close the Graphics Batch.
 - f Save the report to `p13q1.rdf`.
- 2 Reenter the Chart Wizard to change characteristics of the chart created in the previous exercise.
 - a Give the chart the following characteristics:

Title	Order Total per Customer
Chart Type	Column
Chart Subtype	Depth

- b Run the report to test.
 - c Close the Graphics Batch.
 - d Save report to `p13q2.rdf` and close it.

3 Open report `p8q5.rdf`.

- a** Use the Chart Wizard to create a pie chart for the data group `G_PRODUCT`.

Chart Type	Pie
Chart Subtype	Plain
Data Group	G_product
Category	PRODUCT_ID
Value	PROD_TOTAL
Chart Frequency	Once per Name
File Name	p13q3a.ogd

- b** In the Layout editor move the chart beneath the product group and resize it to the same width as the product group.
- c** Run and test the report.
- d** Close the Graphics Batch.
- e** Create a second chart for the customer group.

Chart Type	Pie
Chart Subtype	Plain
Data Group	G_customer
Category	CUSTOMER_ID
Value	CUST_TOTAL
Chart Frequency	Once per Name
File Name	p13q3e.ogd

- f** In the Layout editor move the chart beneath the customer group.
Hint: Create a frame around both charts, so that they are always displayed side by side.
- g** Run and test the report.
- h** Close the Graphics Batch.
- i** Save the report to `p13q3.rdf` and close it.

