Oracle *i*DS Forms: Build Internet Applications I

Volume 1• Instructor Guide

40033GC11

Production 1.1

May 2001

D32775

Author

Ellen Gravina

Technical Contributors and Reviewers

David Ball

Soley Celik

Kathy Cunningham

Ken Goetz

Andrey Ivkin

Kenji Makita

Jayne Marlow

Ian Purvis

Bryan Roberts

Rie Saitoh

Raza Siddiqui

Kajornsak Sungchareon

Ted Witiuk

Publisher

Nita Brozowski

Don Griffin (UK)

Copyright © Oracle Corporation, 2001. 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.

Oracle is a registered trademark and Oracle Forms Developer, Oracle Forms Server, Oracle Reports Developer, Oracle Reports Server, and PL/SQL 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.

Contents

Preface

Introduction

Lesson 1: Introduction to Oracle Forms Developer and Oracle Forms Services

Lesson 2: Running a Form Builder Application

Lesson 3: Working in the Form Builder Environment

Lesson 4: Creating a Basic Form Module

Lesson 7: Creating LOVs and Editors

Lesson 8: Creating Additional Input Items

Lesson 9: Creating Noninput Items

Lesson 10: Creating Windows and Content Canvases

Lesson 11: Working with Other Canvases

Lesson 12: Introduction to Triggers

Lesson 13: Producing Triggers

Lesson 14: Debugging Triggers

Lesson 15: Adding Functionality to Items

Lesson 16: Runform Messages and Alerts

Lesson 17: Query Triggers

Lesson 20: Transaction Processing

Lesson 21: Writing Flexible Code

Lesson 22: Sharing Objects and Code

Lesson 23: Introducing Multiple Form Applications

Appendix A: Practice Solutions

Appendix B: Table Descriptions and Data

Appendix C: Oracle Rdb Overview

Appendix D: Locking in Form Builder

Appendix E: Oracle8 Object Features

Appendix F: Using the Layout Editor

Appendix G: Introduction to Query Builder

Instructor Preface

Instructor Note

General course information for instructors:

PowerPoint Slides The slideshow provides many *builds* to enable you to step through your explanation. Of course, you can modify these builds if you prefer, and if you know how to use PowerPoint.

Demonstrations An additional demonstration, `start.fmb`, is not mentioned in the course instructor notes but is included in the `DEMO` directory. You can deploy this form on the Web and use it to launch all other completed demonstrations provided in the course setup. This form contains a push button for each of the following demonstrations:

- Lesson 4, explicit relationships: `explrela.fmx`
- Lesson 11, windows and canvases: `Win_demo.fmx`
- Lesson 12, execution hierarchy: `EH.fmx`
- Lesson 14, using the debugger: `DebugDemo.fmx`
- Lesson 15, simple hierarchical tree: `HTreeDemo.fmx`
- Lesson 15, hierarchical tree with details: `HTreeDemo2.fmx`
- Lesson 16, generic alert: `Show_alert.fmx`
- Lesson 19, navigation: `navigation.fmx`
- Lesson 21, set item property: `SetFontDemo.fmx`
- Lesson 23, completed application: `customers.fmx`

Practice Sessions Students might want to view the form output that they are asked to produce in each question. Solution files of all forms are available in the `LAB_SOL` directory. These files are named `ORDWK`*XX*`.FMB` and `CUSTWK`*XX*`.FMB`, where *XX* is the lesson number the solution file is for. The solution file for Lesson 1 is called `summitwk.fmb`.

Instructor Note

General setup information for instructors:

Introduction The Oracle Internet Suite is a complete and integrated e-business solution consisting of:

- Oracle8*i* to manage all your content
- Oracle Internet Developer Suite (*i*DS) to build all of your applications
- Oracle9*i* Application Server (9*i*AS) to run all your applications

The Oracle Internet Suite provides integrated development, deployment, and management tools that simplify creating and deploying applications that you need to run your business on the Internet. Oracle Forms Developer, a component of *i*DS, and Oracle Forms Services, a component of 9*i*AS, are key elements in the Oracle Internet Suite. To promote this message, approximately half of the practices for this course were modified for Web deployment. This section provides an overview of the recommended setup for this class with respect to Web deployment.

Recommended Setup The recommended setup for this course implements a three-tier architecture using 9*i*AS as the middle tier application server:

- The client tier contains the Web browser where the application is displayed and used.
- The middle tier is the application server where application logic and server software reside.
- The database tier is the database server where enterprise data is stored.

With this architecture, application logic resides in a single tier and can be maintained easily at one location. The architectural design of the middle tier is optimized for server functions, including access to a database.

Note: Alternative setups are possible. For more information, refer to the manual *Oracle Forms Server Release 6i: Deploying Forms Applications to the Web with Oracle Internet Application Server.*

Java Client With Internet applications, the user interface layer is loaded incrementally on the client in the form of a generic Java applet. This applet is also known as the Java Client or Forms Client. The Java Client has three primary functions:

- To render the Forms Services application display for the user
- To process user interaction back to Forms Services efficiently
- To process incoming messages from Forms Services and translate them into interface objects for the end user efficiently

To run a Java applet, you must install a Java Runtime Environment (JRE) on the client. The JRE may be a part of any Java-enabled Web browser, such as Netscape Navigator or Microsoft Internet Explorer, or it may be installed separately, such as an AppletViewer in a JavaSoft Java Development Kit (JDK). Oracle also offers a special client software product, Oracle JInitiator. JInitiator is an optimized plug-in to a Web browser to run a Forms Services application.

JInitiator is implemented in the setup for this class.

Forms Services Oracle Forms Services, a component of 9*i*AS, deploys Forms applications with database access to Java clients in a Web environment. In addition to the Java Client, the components of Oracle Forms Services and their basic responsibilities are:

- Forms Listener, which acts as a broker, taking connection requests from the client and initiating the run-time process on its behalf. The Listener can maintain a pool of running engines to expedite the connection process.
- Forms Runtime Engine, which handles the user’s context, executing business logic and conducting necessary transactions with the Oracle8*i* database. It performs the same function as the client-server runtime engine, except that all user interface functionality is redirected to the Java client.
- Forms Web Common Gateway Interface (CGI) or Forms Servlet, which dynamically creates an HTML page to be sent back to the user’s browser.

The configuration of Oracle Forms Services and the associated files (`registry.dat` and `formsweb.cfg`) are not covered in this course. There are notes throughout the course that refer students to the course *Oracle 9iAS Forms Services: Deploy Internet Applications* for more information.

Web Listener In order to run Forms applications on the Web, a Web listener is required on the middle tier. The Web listener services requests from the client to run a Forms application.

Oracle9*i* Application Server uses Oracle HTTP Server, powered by Apache Web server technology, to provide Web listening services.

Steps to Deploy Forms Applications on the Web

1. Configure the Oracle HTTP Server.

   This is done for you in the installation of 9*i*AS.

2. Configure the Forms Services environment variables.

   This is done for you in the installation of 9*i*AS.

Steps to Deploy Forms Applications on the Web (continued)

As part of the course setup, the FORMS60_PATH variable in the registry of the middle-tier application server should be set to the directory that will contain the executable Forms applications (the FMX files).

3 Verify that the Oracle HTTP Server is running.

– Open Netscape Navigator.

– In the Location field, enter the URL of the middle-tier application server. This information is specific to each site. For example, if 9*i*AS was installed on the machine identified as cbauwens-lap, you would enter `http://cbauwens-lap`.

A page titled Oracle HTTP Server Components should be returned to your browser window.

4 Verify that Forms Services is running.

– In the Location field of your browser, enter the following: `http://<middle-tier application server name>/dev60cgi/ifcgi60.exe?`

Following the example above, you would enter: `http://cbauwens-lap/dev60cgi/ifcgi60.exe?`

A window should display in your browser stating Oracle Forms Server is successfully installed.

5 Generate the executable (FMX) for the Forms application you want to deploy on the Web.

6 Deploy the Forms application on the Web. Double-click the shortcut on your desktop for Run a Form on the Web. If there is no shortcut, you can invoke the Web Forms Tester from the Windows NT task bar:

– Click the Start button and select Programs.

– Select Oracle Forms 6*i*.

– Select Run a Form on the Web.

In the Web host field, specify the name of the middle-tier machine where Forms Services and the Oracle HTTP Server are running. In the Web port field, specify a port number for the Oracle HTTP Server. This information is specific to each site.

You can create a bookmark for this location:

– Click the Bookmarks button.

– Select Add Bookmark.

Steps to Deploy Forms Applications on the Web (continued)

Run a form by entering the name of your executable file for the Form parameter and the information to connect to the database for the Userid parameter. Click the Run Form button.

Note: The Run Form Web button in Form Builder will not be functional if the recommended setup for this class is implemented. According to the release notes for Forms Developer 6*i*, Forms Services must be installed on the client machine in order for this button to be operational.

Preface

Profile

Before You Begin This Course

Before you begin this course, you should have the following qualifications:

- 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*
  - *Advanced SQL and SQL*Plus*
  - *Develop PL/SQL Program Units*
- Or the following CBTs from our SQL * PL/SQL CBT Library:
  - *Oracle SQL and SQL*Plus: Basic SELECT Statements* or *Oracle SQL Specifics: Retrieving and Formatting Data*
  - *Oracle SQL and SQL*Plus: DDL and DML* or *Oracle SQL Specifics: Creating and Managing Database Objects*
  - *Oracle PL/SQL: Basics*
  - *Oracle SQL and SQL*Plus: Advanced SELECT Statements*
  - *Oracle SQL and SQL*Plus: SQL*Plus and Reporting*
  - *Oracle PL/SQL: Procedures, Functions and Packages*
  - *Oracle PL/SQL: Database Programming*

Suggested Follow-up Courses

- *Oracle iDS Forms: Build Internet Applications II*
- *Oracle iDS Reports: Build Internet Reports*
- *Oracle 9iAS Forms Services: Deploy Internet Applications*

How This Course Is Organized

Oracle iDS Forms: Build Internet Applications I 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**
Oracle Forms Developer Release 6i: Getting Started (Windows 95/NT)	A73154-01
Oracle Forms Developer and Reports Developer Release 6i: Guidelines for Building Applications	A73073-02
Oracle Forms Server Release 6i Patch 2: Deploying Forms Applications to the Web with the Oracle Internet Application Server	A86202-01

Additional Publications

- `read.me` files
- `relnotes.pdf` file

Typographic Conventions

Typographic Conventions in Text

Convention	**Element**	**Example**
Bold italic	Glossary term (if there is a glossary)	The ***algorithm*** 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: `debug.seti('I',300);` Directory: `bin` (DOS), `$FMHOME` (UNIX) Filename: Locate the `init.ora` file. Password: Use `tiger` as your password. Pathname: Open `c:\my_docs\projects` URL: Go to `http://www.oracle.com` User input: Enter `300` Username: Log on as `scott`
Initial cap	Graphics labels (unless the term is a proper noun)	Customer address (*but* Oracle Payables)
Italic	Emphasized words and phrases, titles of books and courses, variables	Do *not* save changes to the database. For further information, see *Oracle7 Server SQL Language Reference Manual.* Enter *`user_id`*`@us.oracle.com`, where *user_id* 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` *`role`*

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

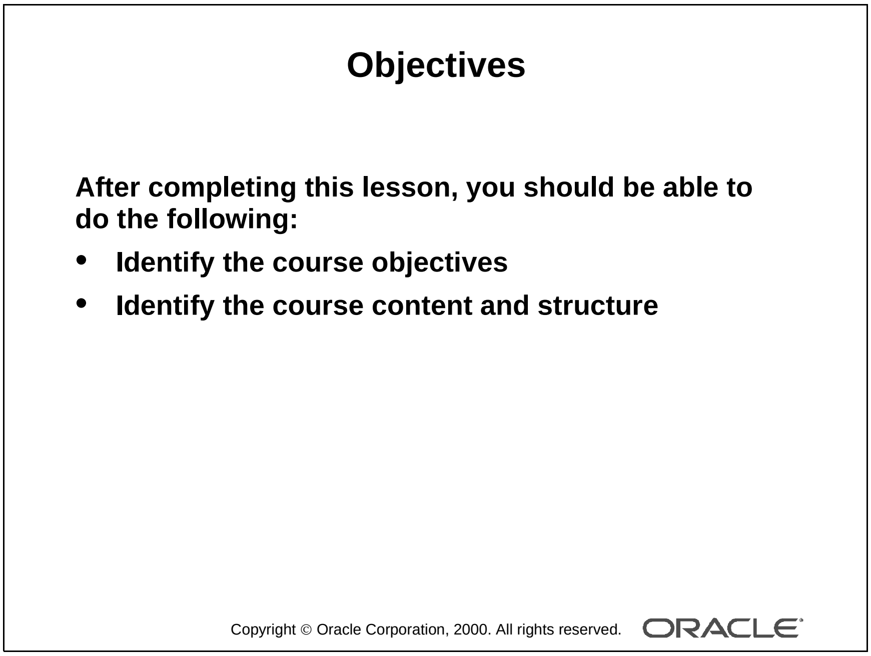
This simplified path translates to the following:

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

After completing this lesson, you should be able to do the following:

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

Copyright © Oracle Corporation, 2000. All rights reserved.

Schedule

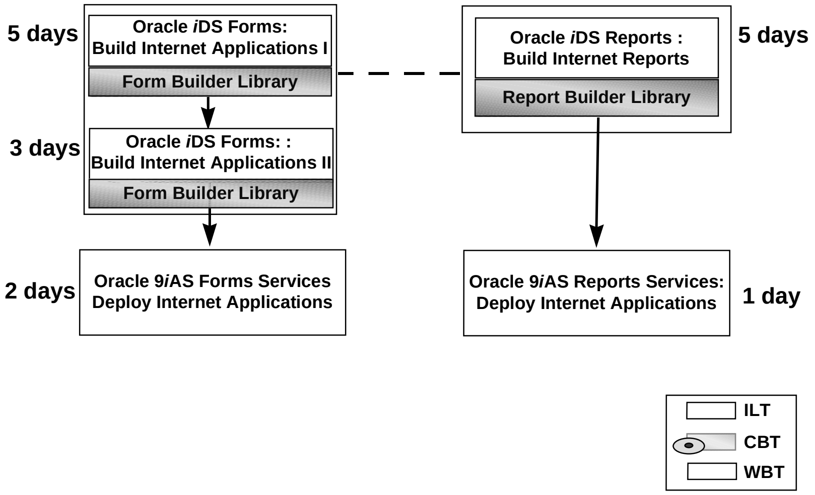
Topic	**Timing**
Lecture	15 minutes
Total	15 minutes

Overview

Introduction

This lesson introduces you to the *Oracle 9iDS Forms: Build Internet Applications I* 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.

Release 6*i* Curriculum



5 days Oracle *i*DS Forms: Build Internet Applications I

Form Builder Library

3 days Oracle *i*DS Forms: : Build Internet Applications II

Form Builder Library

2 days Oracle 9*i*AS Forms Services Deploy Internet Applications

Oracle *i*DS Reports : Build Internet Reports 5 days

Report Builder Library

Oracle 9*i*AS Reports Services: Deploy Internet Applications 1 day

Copyright © Oracle Corporation, 2000. All rights reserved.

Course Objectives

After completing this course, you should be able to do the following:

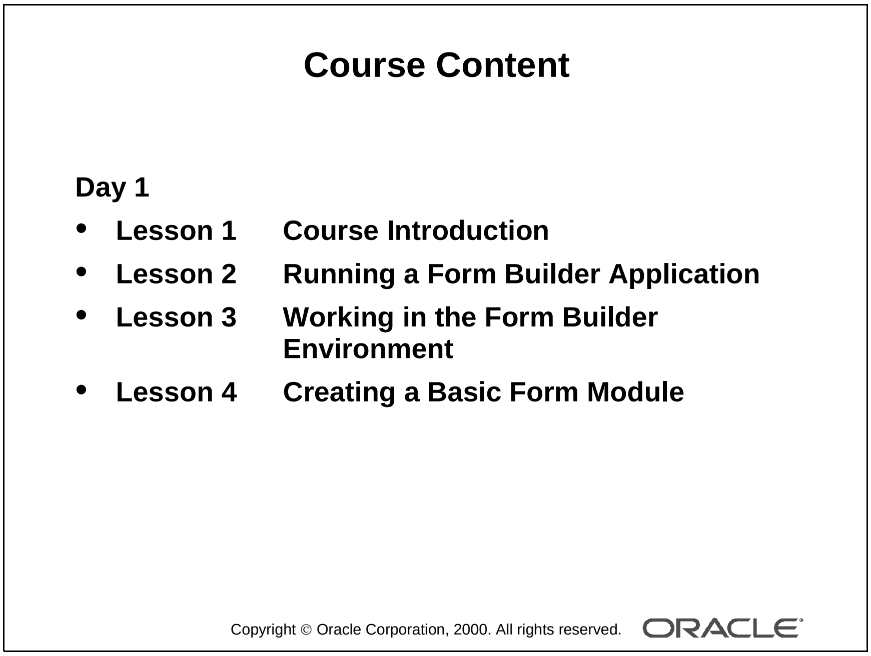
- Describe the capabilities of Oracle Forms Developer
- Create form modules including components for database interaction and GUI controls
- Display form modules in multiple windows and a variety of layout styles
- Implement triggers to:
  - Enhance functionality
  - Communicate with users
  - Supplement validation, control navigation, and modify default transaction processing
- Reuse objects and code

Copyright © Oracle Corporation, 2000. All rights reserved.

Course Objectives

Course Description

In this course, participants will build, test, and deploy interactive Internet applications. Working in a graphical user interface (GUI) environment, participants will learn how to create and customize forms with user input items such as check boxes, list items, and radio groups. They will also learn how to modify data access by creating event-related triggers.

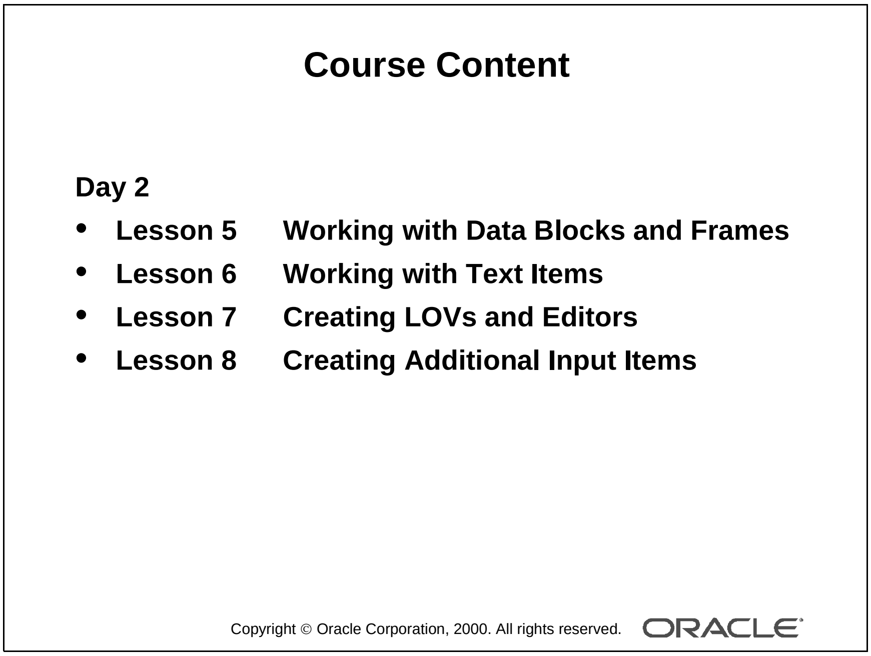


Course Content

Day 1

- **Lesson 1 Course Introduction**
- **Lesson 2 Running a Form Builder Application**
- **Lesson 3 Working in the Form Builder Environment**
- **Lesson 4 Creating a Basic Form Module**

Copyright © Oracle Corporation, 2000. All rights reserved.



Course Content

Day 2

- **Lesson 5 Working with Data Blocks and Frames**
- **Lesson 6 Working with Text Items**
- **Lesson 7 Creating LOVs and Editors**
- **Lesson 8 Creating Additional Input Items**

Copyright © Oracle Corporation, 2000. All rights reserved.

Course Content

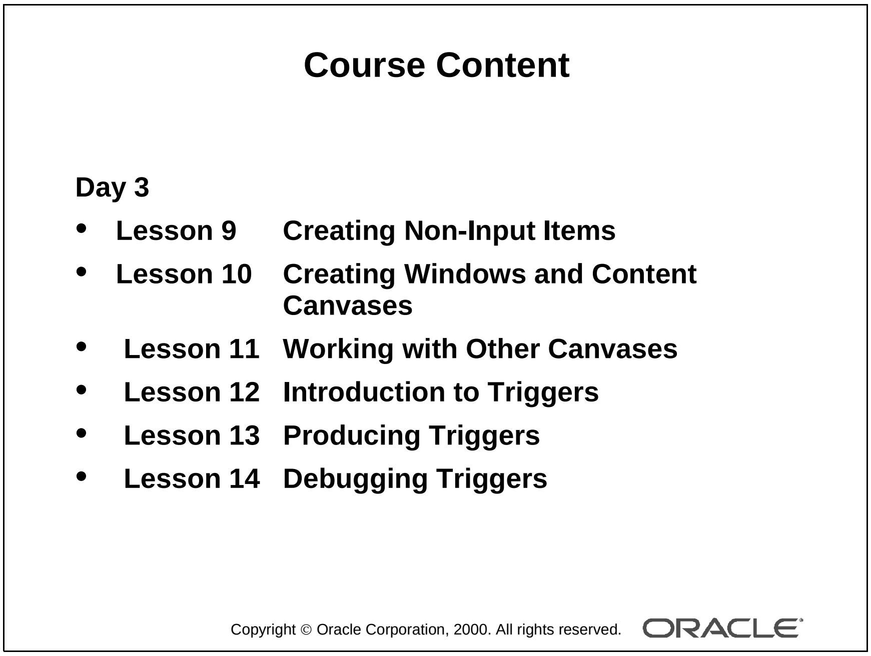
The lesson titles show the topics we cover in this course, and the usual sequence of lessons. However, the daily schedule is an estimate, and may vary for each class.

Day 1

Lesson Number	**Lesson Name**
1	Course Introduction
2	Running a Form Builder Application
3	Working in the Form Builder Environment
4	Creating a Basic Form Module

Day 2

Lesson Number	**Lesson Name**
5	Working with Data Blocks and Frames
6	Working with Text Items
7	Creating LOVs and Editors
8	Creating Additional Input Items

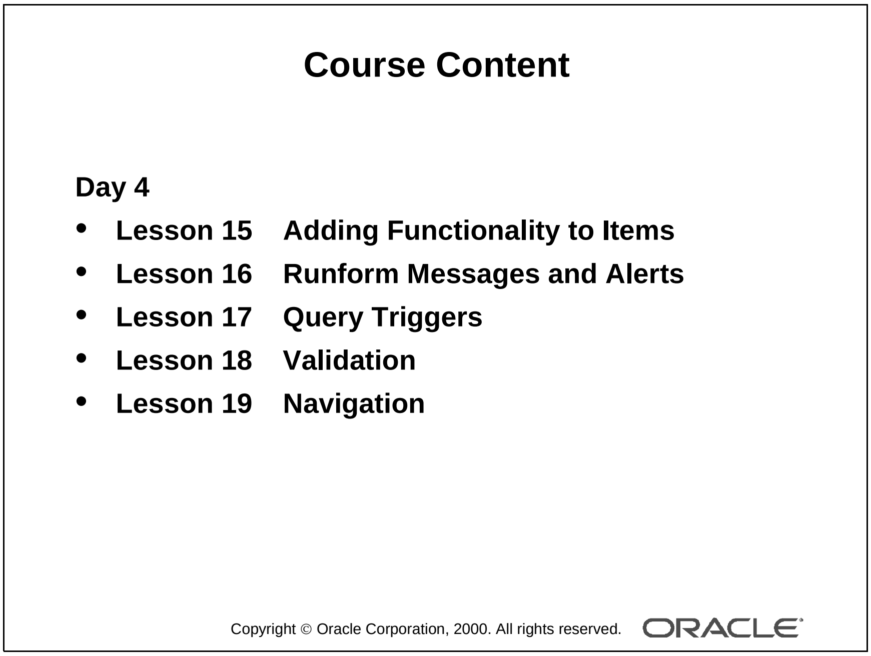


Course Content

Day 3

- **Lesson 9 Creating Non-Input Items**
- **Lesson 10 Creating Windows and Content Canvases**
- **Lesson 11 Working with Other Canvases**
- **Lesson 12 Introduction to Triggers**
- **Lesson 13 Producing Triggers**
- **Lesson 14 Debugging Triggers**

Copyright © Oracle Corporation, 2000. All rights reserved.



Course Content

Day 4

- **Lesson 15 Adding Functionality to Items**
- **Lesson 16 Runform Messages and Alerts**
- **Lesson 17 Query Triggers**
- **Lesson 18 Validation**
- **Lesson 19 Navigation**

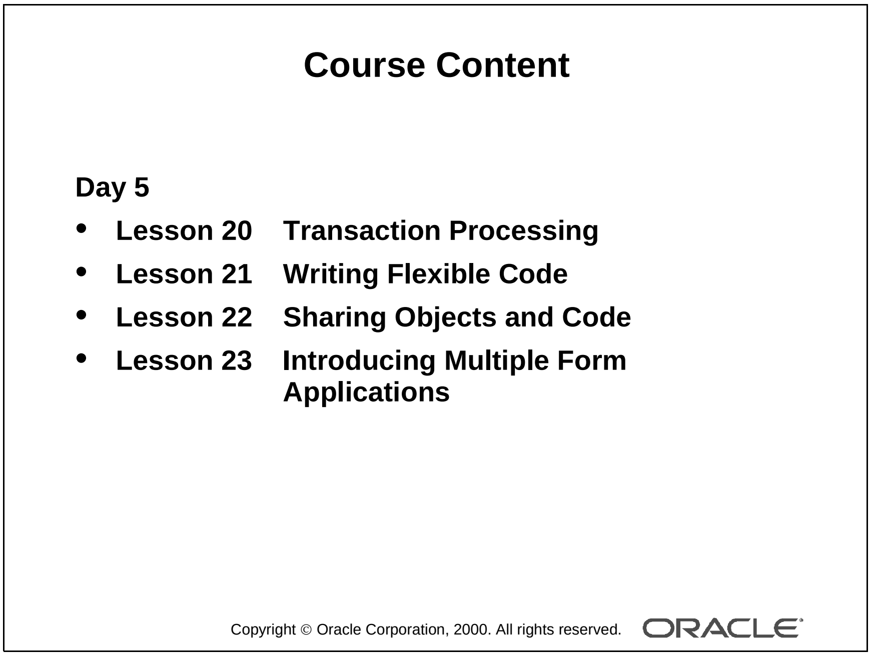
Copyright © Oracle Corporation, 2000. All rights reserved.

Day 3

Lesson Number	Lesson Name
9	Creating Non-Input Items
10	Creating Windows and Content Canvases
11	Working with Other Canvases
12	Introduction to Triggers
13	Producing Triggers
14	Debugging Triggers

Day 4

Lesson Number	Lesson Name
15	Adding Functionality to Items
16	Runform Messages and Alerts
17	Query Triggers
18	Validation
19	Navigation



Course Content

Day 5

- **Lesson 20 Transaction Processing**
- **Lesson 21 Writing Flexible Code**
- **Lesson 22 Sharing Objects and Code**
- **Lesson 23 Introducing Multiple Form Applications**

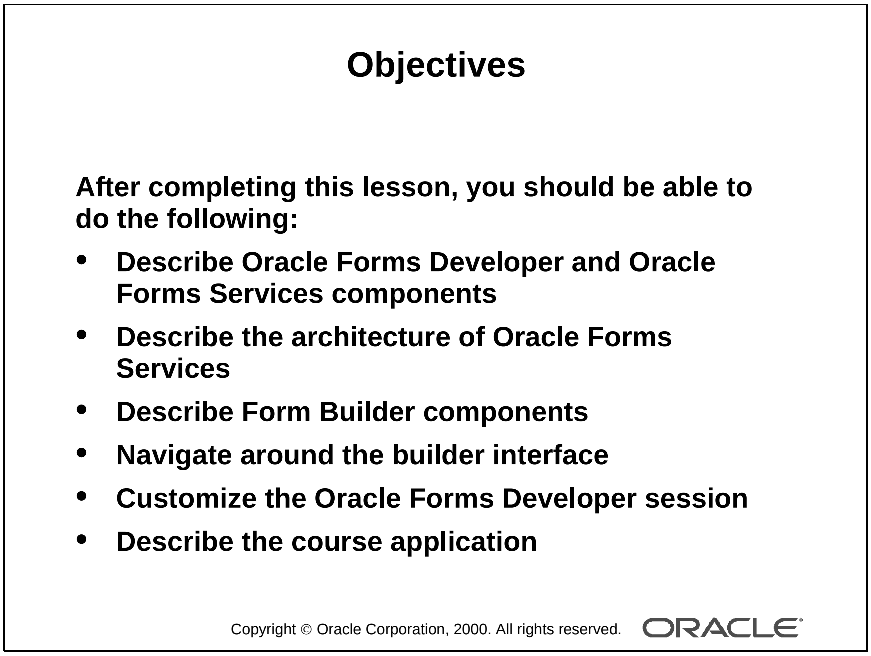
Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Day 5

Lesson Number	Lesson Name
20	Transaction Processing
21	Writing Flexible Code
22	Sharing Objects and Code
23	Introducing Multiple Form Applications

1

Introduction to Oracle Forms Developer and Oracle Forms Services



Objectives

After completing this lesson, you should be able to do the following:

- **Describe Oracle Forms Developer and Oracle Forms Services components**
- **Describe the architecture of Oracle Forms Services**
- **Describe Form Builder components**
- **Navigate around the builder interface**
- **Customize the Oracle Forms Developer session**
- **Describe the course application**

Copyright © Oracle Corporation, 2000. All rights reserved.

Schedule

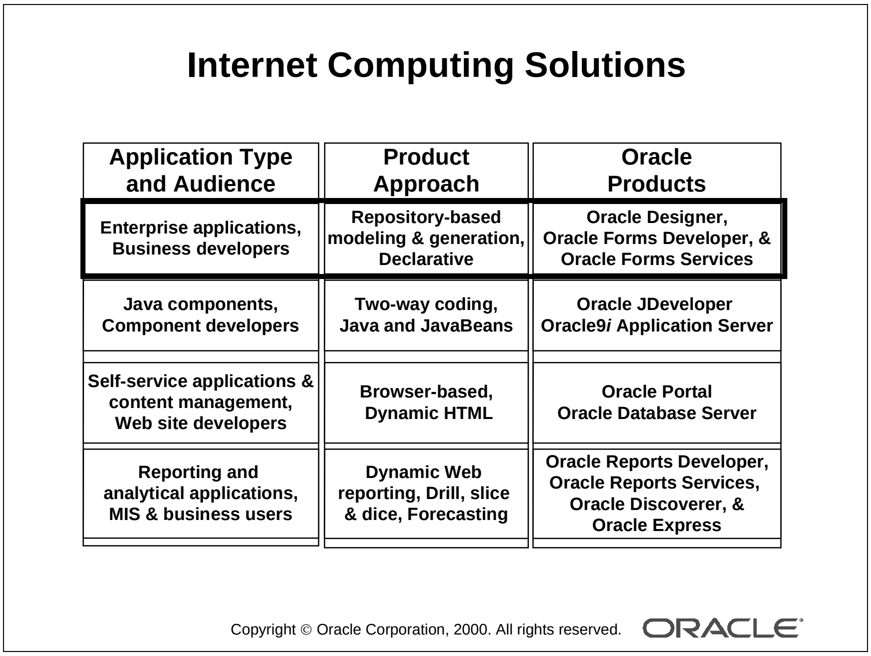
Topic	**Timing**
Lecture	30 minutes
Practice	25 minutes
Total	55 minutes

Introduction

Overview

This course teaches you how to build effective and professional form applications using Oracle Forms Developer.

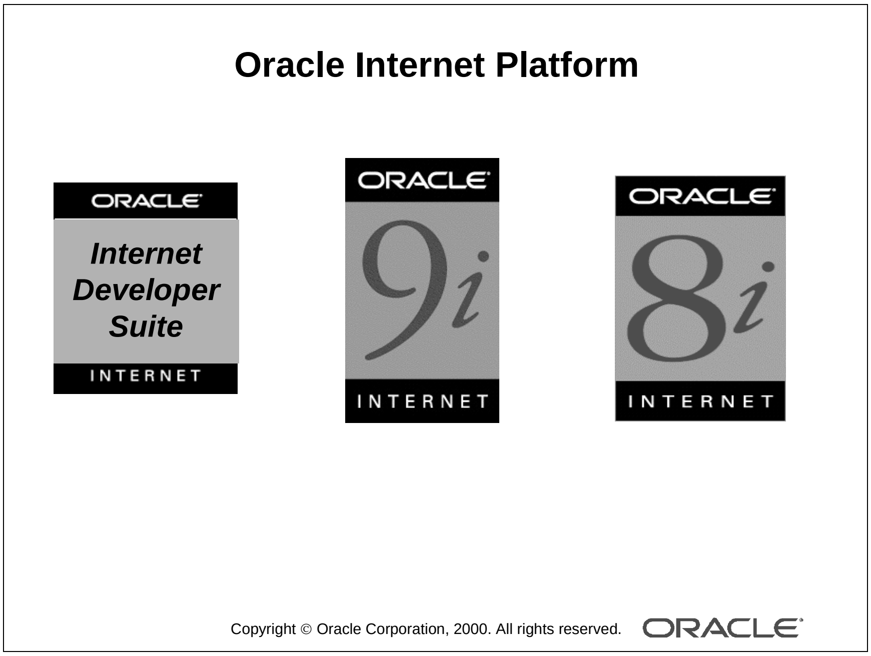
This lesson identifies the key features of Oracle Forms Developer, Oracle Forms Services, and the course application model and contents.



Internet Computing Solutions

Application Type and Audience	Product Approach	Oracle Products
Enterprise applications, Business developers	Repository-based modeling & generation, Declarative	Oracle Designer, Oracle Forms Developer, & Oracle Forms Services
Java components, Component developers	Two-way coding, Java and JavaBeans	Oracle JDeveloper Oracle9*i* Application Server
Self-service applications & content management, Web site developers	Browser-based, Dynamic HTML	Oracle Portal Oracle Database Server
Reporting and analytical applications, MIS & business users	Dynamic Web reporting, Drill, slice & dice, Forecasting	Oracle Reports Developer, Oracle Reports Services, Oracle Discoverer, & Oracle Express

Copyright © Oracle Corporation, 2000. All rights reserved.



Oracle Internet Platform

Copyright © Oracle Corporation, 2000. All rights reserved.

Internet Computing Solutions

Oracle offers a range of tools and deployment options for Internet computing. Different types of developers and different types of applications require different toolsets.

Enterprise application developers need a declarative model-based approach. Oracle Designer and Oracle Forms Developer provide this solution, using Oracle Forms Services as the primary deployment option.

Component developers need different tools and methods. For these developers, Java is the language of choice. Oracle’s solution is JDeveloper.

For Web site developers and content publishers who want to build self-service, dynamic Hypertext Markup Language (HTML) applications for Web sites, the approach is different. Oracle Portal provides an easy-to-use development environment that resides entirely inside an Oracle8*i* database. Portal provides a browser-based environment from development through deployment of an application.

For Management Information System (MIS) developers and end users, there is the Oracle Business Intelligence toolset. Oracle Reports Developer, Oracle Reports Services, Oracle Discoverer, and Oracle Express provide the whole range for reporting, analysis, and trending facilities.

This course focuses on how you can rapidly build scalable, high-performance applications for the Internet by using Oracle Forms Developer.

Oracle Internet Platform

Recently, Oracle has revamped its infrastructure software to make it easier for companies interested in e-commerce to create scalable, Web-based applications.

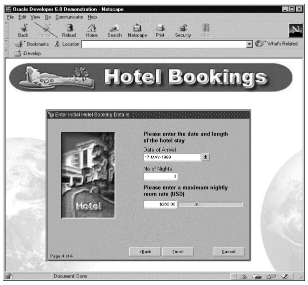
The Oracle Internet Developer Suite (*i*DS) brings together in a single package several of Oracle’s development tools, specifically, Oracle Forms Developer, Oracle Designer, Oracle Reports, Oracle JDeveloper, Oracle Portal, and Oracle Discoverer.

The Oracle9*i* Application Server (9*i*AS) is a Java-based application server that provides deployment and management tools to simplify the deployment of your e-business applications that need to run on the Internet. Coupled with the Oracle8*i* database, Oracle9*i* Application Server provides a reliable, scalable, and secure middle-tier application server.

What Is Oracle Forms Developer?

A productive development environment for Internet business applications

- **Data entry**
- **Query screens**



Oracle Developer 6.0 Demonstration - Netscape

Hotel Bookings

Enter Initial Hotel Booking Details

Please enter the date and length of the hotel stay

Date of Arrival: 17-MAY-1999

No of Nights: 3

Please enter a maximum nightly room rate (USD)

$260.00

Page 4 of 4

Document: Done

Copyright © Oracle Corporation, 2000. All rights reserved.

What Is Oracle Forms Developer?

Oracle Forms Developer, a component of the Oracle Internet Developer Suite, is a productive development environment for building enterprise-class, scalable database applications for the Internet. Oracle Forms Developer provides a set of tools that enable business developers to easily and quickly construct sophisticated database forms and business logic with a minimum of effort.

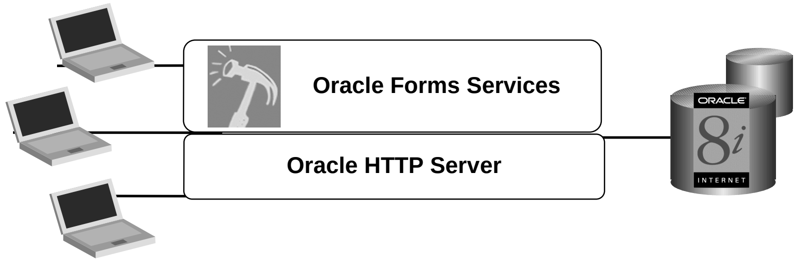
Oracle Forms Developer uses powerful declarative capabilities to rapidly create applications from database definitions that leverage the tight integration with Oracle8*i*. The toolset leverages Java technology, promotes reuse, and is designed to allow developers to declaratively build rich user interfaces.

Developer productivity is further increased through a single integrated development environment that enables distributed debugging across all tiers, utilizing the same PL/SQL language for both server and client.

Oracle Forms Developer's tight integration with Oracle Designer enables you to use a productive model-driven development approach. Oracle Forms Developer applications can be automatically generated from business requirements designed in the Oracle Designer modeling environment. These models are stored in the Oracle Repository. Code-level changes made within the Oracle Forms Developer environment can be automatically reverse engineered back into the models, preserving the integrity between the models and the application.

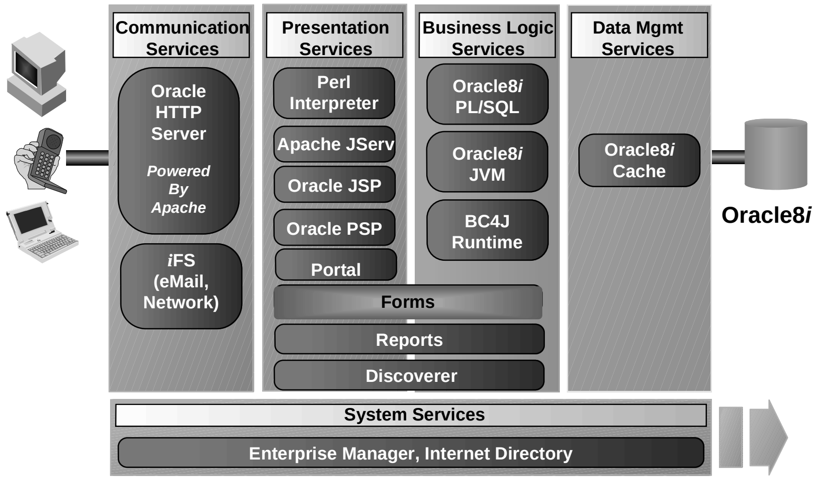
What Is Oracle Forms Services?

A component of Oracle9*i* Application Server that deploys Forms applications to Java clients in a Web environment



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

9*i*AS Major Elements



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

What Is Oracle Forms Services?

Oracle Forms Services is a component of Oracle9*i* Application Server for delivering Oracle Forms Developer applications to the Internet. Oracle Forms Services automatically provides the infrastructure needed to successfully deliver applications on the Internet through built-in services and optimizations.

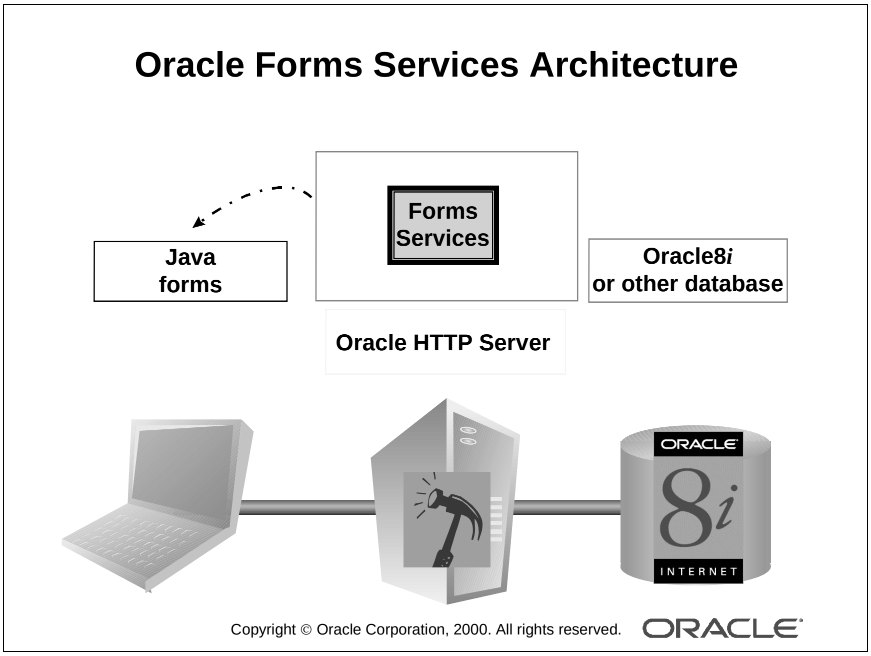
Oracle Forms Services uses a three-tier architecture to deploy database applications:

- The client tier contains the Web browser, where the application is displayed and used.
- The middle tier is the application server, where the application logic and server software reside.
- The database tier is the database server, where enterprise data is stored.

Other Major Elements in Oracle9*i* Application Server

9*i*AS consists of a set of services and utilities that can be used to implement applications in a distributed environment for scalability and reliability. They are:

- Communication Services: These services handle incoming requests received by 9*i*AS. Some of these requests are processed by the Oracle HTTP Server and some requests are routed to other areas of 9*i*AS for processing.
- Presentation Services: The presentation services of 9*i*AS generally output some kind of graphical representation, often in the form of HTML.
- Business Logic Services: 9*i*AS provides several ways to develop business logic, utilizing both Java development approaches and high-level model-driven techniques. These approaches include Java technologies such as Enterprise JavaBeans (EJB) and Oracle Business Components for Java (BC4J), as well as rich GUI oriented approaches such as Oracle Forms and Reports.
- Data Management Services: To reduce the load on the back-end database instance, and to avoid network roundtrips for read-only data, 9*i*AS includes Oracle8*i* cache.
- System Services: To provide system management and security services, 9*i*AS includes Oracle Enterprise Manager and Oracle Advanced Security. These system services provide a comprehensive management framework for your entire Oracle environment and network security using Secure Sockets Layer (SSL)-based encryption and authentication facilities.



Oracle Forms Services Architecture

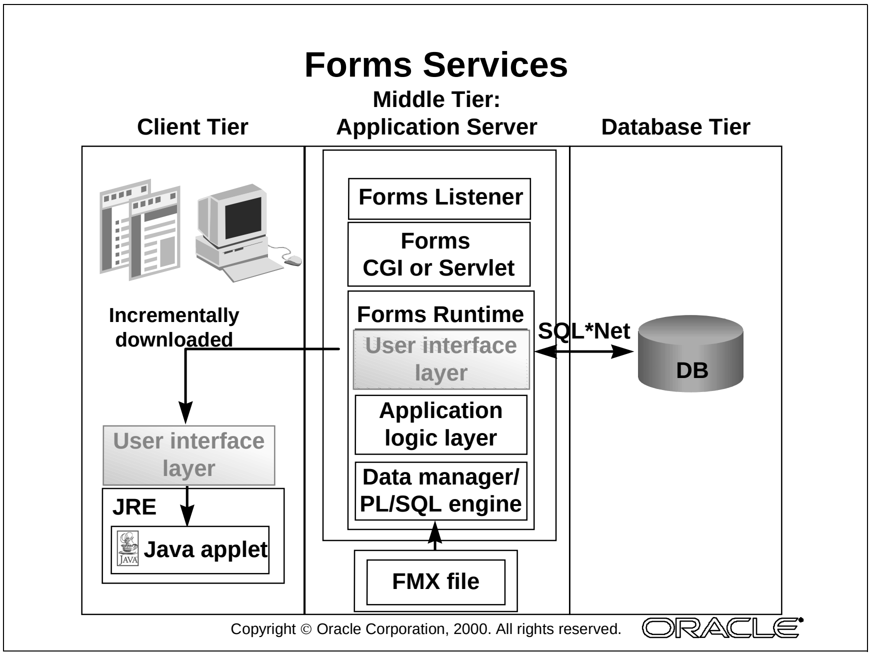
Java forms

Forms Services

Oracle HTTP Server

Oracle8*i* or other database

Copyright © Oracle Corporation, 2000. All rights reserved.



Forms Services

Client Tier

Middle Tier: Application Server

Database Tier

Incrementally downloaded

User interface layer

JRE

Java applet

Forms Listener

Forms CGI or Servlet

Forms Runtime

User interface layer

Application logic layer

Data manager/ PL/SQL engine

FMX file

SQL*Net

DB

Copyright © Oracle Corporation, 2000. All rights reserved.

Forms Services Architecture

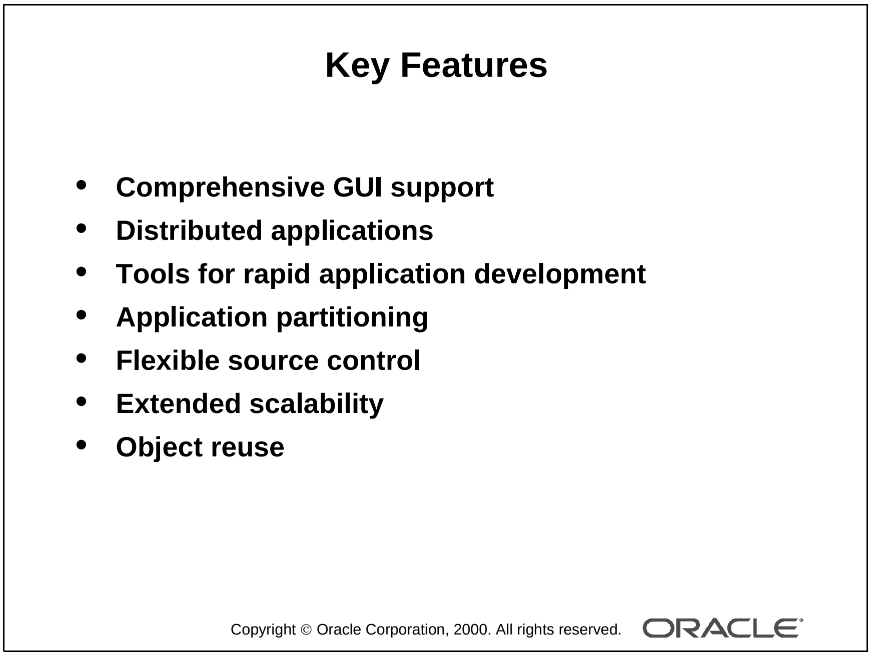
Oracle Forms Services, a component of Oracle 9*i*AS, is an application service for deploying complex, transactional forms applications to the Internet. Forms Services consists of four major components: the Java client, the Forms Listener, the Forms Common Gateway Interface (CGI) or Servlet, and the Forms Runtime Engine.

- Java client: The Java client is an applet that is downloaded at run time from an application server to an end user’s Web browser. The Java client displays the forms’s user interface and manages interaction between end users and Forms Services.
- Forms Listener: The Forms Listener initiates the Forms Services run-time session and establishes a connection between the Forms client and the Forms Services Runtime Engine. The listener can also maintain a pool of run-time engines ready for connection, making connection time from the Java client complete as quickly as possible.
- Forms CGI or Servlet: The Forms CGI or Servlet dynamically creates an HTML page to be sent back to the user’s browser, provides an optional entry point to Forms applications, and administers load balancing.
- Forms Runtime Engine: The Forms Runtime Engine manages application logic and application processing. It is also the process that maintains a connection to the database on behalf of the Java client. The code run by the Forms Services Runtime Engine is the same code—form, menu, and library modules—used for running in a client-server implementation on the same platform. No application code changes are required to deploy an application to the Internet.

When a user runs a forms session over the Web, a thin, Java-based Forms applet is dynamically downloaded from the application server and automatically cached on the Java client machine. The same Java applet code can be used for any form, regardless of size and complexity.

Instructor Note

See *Deploying Forms Applications to the Web with Oracle Internet Application Server* for additional information on Forms Services.



Key Features

- Comprehensive GUI support
- Distributed applications
- Tools for rapid application development
- Application partitioning
- Flexible source control
- Extended scalability
- Object reuse

Copyright © Oracle Corporation, 2000. All rights reserved.

Oracle Forms Developer Key Features

Oracle Forms Developer provides a number of features that contribute to the strength and flexibility of the product.

Comprehensive GUI Support Oracle Forms Developer supports the native features of Microsoft Windows 95 and NT 4.0, and it provides portability to Motif and character-mode production environments.

Distributed Applications The tools provide local, client-server, and Web support with multiple database connections per application. In addition to Oracle7 and Oracle8, your applications can access Structured Query Language (SQL) databases through open database connectivity (ODBC).

Tools for Rapid Application Development Oracle Forms Developer provides the tools that simplify many development tasks, enabling you to create and modify applications with little or no code. Productivity is enhanced with Oracle Forms Developer's wizard-based rapid application development and built-in commands that perform common functions.

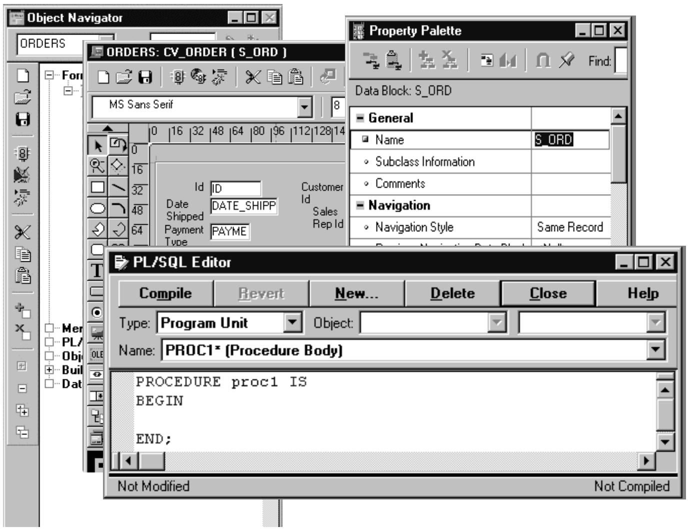
Application Partitioning You can place individual PL/SQL program units on the database server, the application server, or in the client-side application, whichever is most suitable in each case. You can copy and move objects between modules and the database server by using convenient drag-and-drop techniques.

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 by using Oracle Forms 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 Reuse Oracle Forms 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.

Form Builder Components



Data Block: S_ORD

General	
Name	S_ORD
Subclass Information	
Comments	
Navigation	
Navigation Style	Same Record

Id: ID
Date Shipped: DATE_SHIPP
Payment Type: PAYME
Customer Id
Sales Rep Id

Type: Program Unit
Object:
Name: PROC1* (Procedure Body)

```
PROCEDURE proc1 IS
BEGIN

END;
```

Not Modified
Not Compiled

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Mention these briefly. We will start Form Builder and see the interface in a demonstration later in this course.

Form Builder Components

Introduction

Form Builder is the main application-building component of Oracle Forms Developer.

The interface components of Form Builder help to provide the flexibility and productivity of the Oracle Forms Developer development environment.

Object Navigator

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

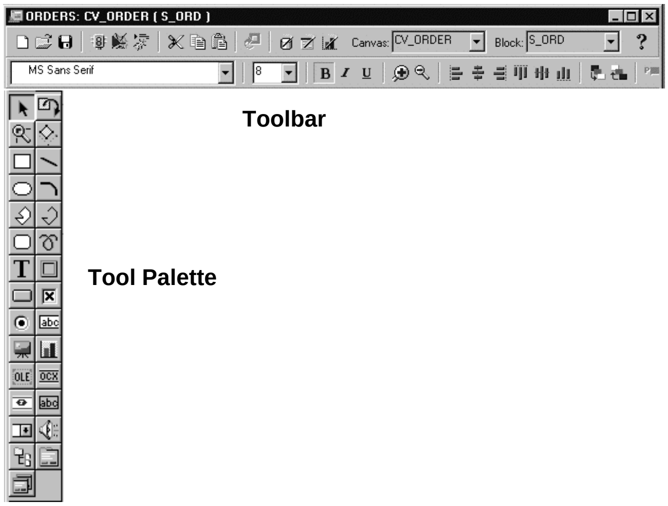
- A hierarchy represented by indentation and expandable nodes. (Top-level nodes show module types, database objects, and built-in packages. All other nodes and the objects they contain are indented to indicate that they belong to these higher level nodes.)
- Find field and icons, enabling forward and 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.
- An icon next to each object to indicate the object type.

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

Form Builder Layout Editor



Canvas: CV_ORDER

Block: S_ORD

Toolbar

Tool Palette

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

It is important for the students to gain exposure to the Object Navigator and the Layout Editor in Form Builder. They will need this knowledge to complete the practice exercise at the end of this lesson.

Demonstration

- Open an existing form, and briefly explain the features of the Object Navigator.
- Go to the Layout Editor. Point out the major tools in the Toolbar. Demonstrate some of the major tools in the Tool Palette.
- Draw a rectangle and an ellipse, and also show how to draw a square and a circle by holding down the [Shift] key when using the rectangle and ellipse tools.
- Demonstrate selecting more than one object at the same time, and show how to group various objects.
- Demonstrate resizing, moving, aligning, coloring, and deleting objects.
- Demonstrate using the Text tool and changing the font style and size.
- Demonstrate how to pin a tool by double-clicking a tool.

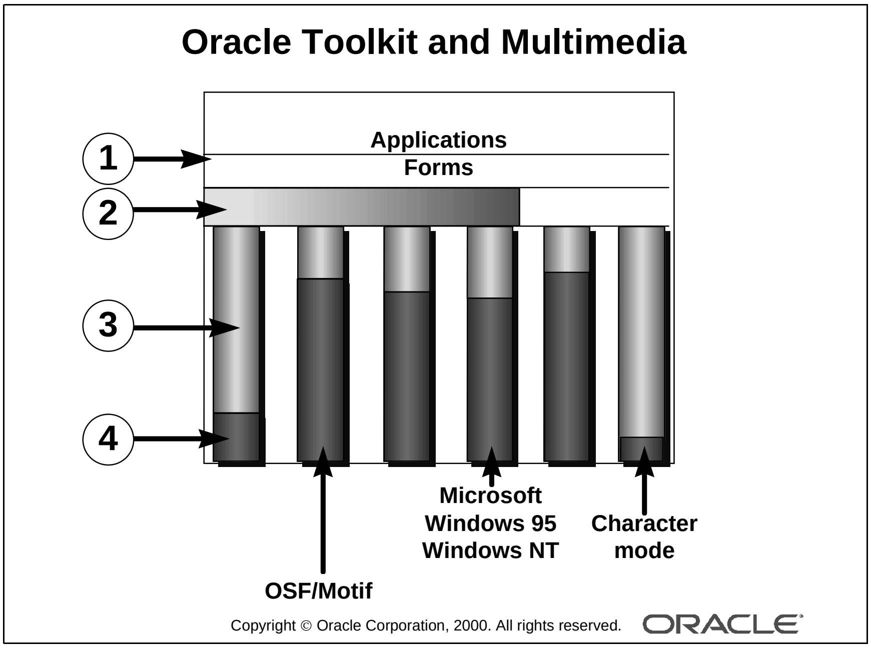
Layout Editor (or Layout Model)

The Layout Editor is a graphical design facility for creating and arranging interface items and graphical objects in your application. When you use the Tool Palette and the Toolbar available in the Layout Editor, you can design the style, color, size, and arrangement of visual objects in the application. The layout can include graphical objects and bitmapped images when running in a GUI environment.

PL/SQL Editor

The PL/SQL Editor is the integrated functionality of Oracle Procedure Builder that exists within the Form Builder. It provides:

- Development of triggers, procedures, functions, and packages in Oracle Forms Developer as well as the database
- Development of libraries to hold PL/SQL program units
- Statement-level debugging of PL/SQL at run time



Oracle Toolkit and Multimedia

Copyright © Oracle Corporation, 2000. All rights reserved.

1	Oracle Forms Developer
2	Multimedia extension
3	Toolkit
4	Base functionality

Instructor Note

ActiveX controls are reusable components that can be embedded in a Form Builder application. Form Builder acts as an ActiveX control container.

The course *Oracle 9iDS Forms: Build Internet Applications II* discusses the use of ActiveX controls.

Oracle Toolkit and Oracle Multimedia

Oracle Toolkit is a library of functions that perform user interface events, such as control of the scroll bar and menu activation.

Together with Oracle Multimedia, which provides the integration of images, sounds, and other media facilities, Toolkit underlays Oracle Forms Developer, bridging the gap between your Oracle Forms Developer applications and the native environment in which you are working.

As your applications request facilities, such as opening a window or displaying a menu, Oracle Forms Developer passes the requests to Toolkit, which communicates them to the native platform.

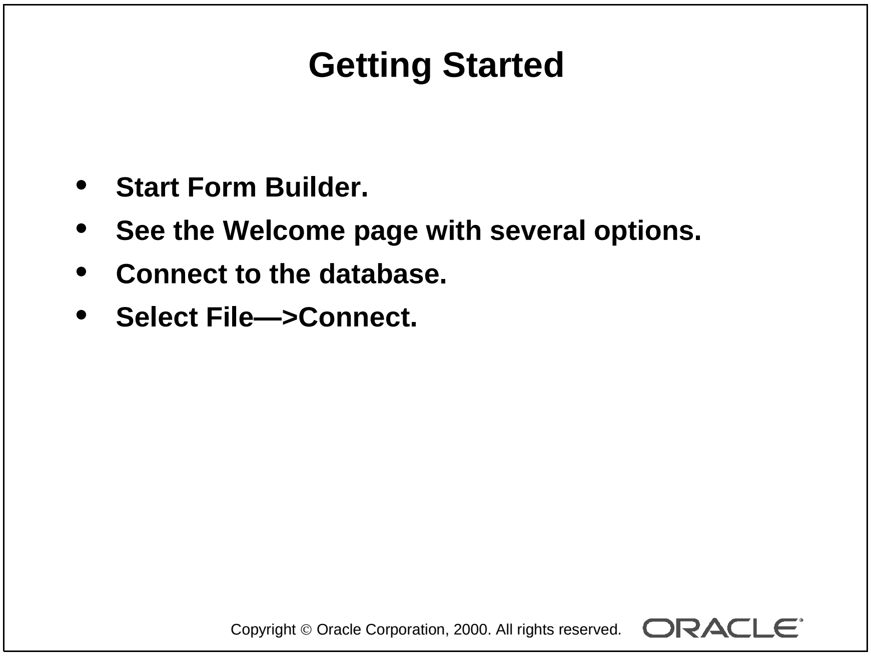
Toolkit attempts to pass tasks to the native interface, if that interface can handle them, so that your application uses the natural features of your environment, wherever possible. This means that an Oracle Forms Developer application looks and behaves like a Windows NT application when running on Windows NT, and like a Motif application if moved to this platform.

Toolkit itself provides the functionality where certain functionalities are not available through the native interface, for example, on character mode devices. This provides an adaptable user interface for your Oracle Forms Developer applications.

Note: You can see a list of these and other products that support the common components and integration of Oracle Forms Developer by selecting Help—>About in the Form Builder menu.

Web Design Tip

ActiveX, OCX, OLE, and VBX: Third party controls that display screen output on the application server are not supported in a form deployed to the Web because users cannot view the output. Equivalent functionality can be obtained by using JavaBeans and pluggable Java components (PJCs) in Internet applications.



Getting Started

- **Start Form Builder.**
- **See the Welcome page with several options.**
- **Connect to the database.**
- **Select File—>Connect.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Invoke Form Builder.

Open an existing form. Use one of the forms from the existing prebuilt application.

We use an existing form in this lesson to familiarize students with the Oracle Forms Developer interface. Later we create new forms by using the wizards.

Form Builder will probably prompt you to connect to the database when it tries to run the form. Also show students the File—>Connect menu option.

Getting Started in the Oracle Forms Developer Interface

Starting Form Builder

To start Form Builder, invoke it from the Forms Developer 6*i* group.

What You See in the Builders

When you invoke Form Builder, you first see the Welcome page giving you several options, including:

- Create a new module by using the Data Block 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 Form Builder.

Connect to a database if you need to:

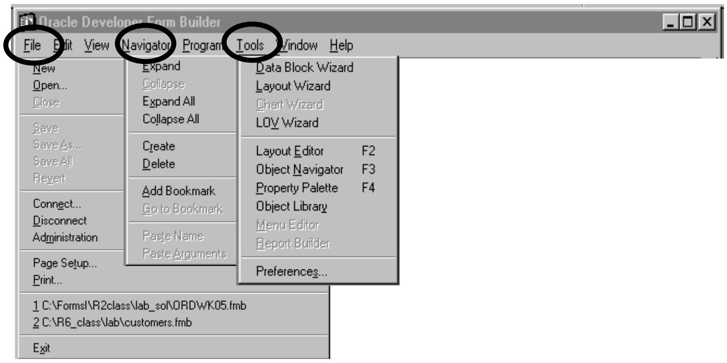
- Compile code that contains SQL
- Access database objects in the Object Navigator
- Create Oracle Forms Developer objects that are based on database objects

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 not connecting to the default database, also provide the necessary connect string or database alias.

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

Main Menu Structure



Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Show the Main menu options, and how they differ in context.

Switch between Object Navigator and Layout Editor. The Navigator menu option is replaced with the Format and Arrange options while in the Layout Editor.

Do not describe Help in detail; there are more details for these options later in this lesson.

Navigating Around the Form Builder Main Menu

Introduction

The Main menu contains options to enable you to create, modify, and manage your form modules.

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 using a dialog box
>	Menu option has a submenu
Windows menu	List of open windows; select any window to make it active
Help	List of help facilities, such as online help text, Cue Cards, Quick Tour

Native GUI Interface

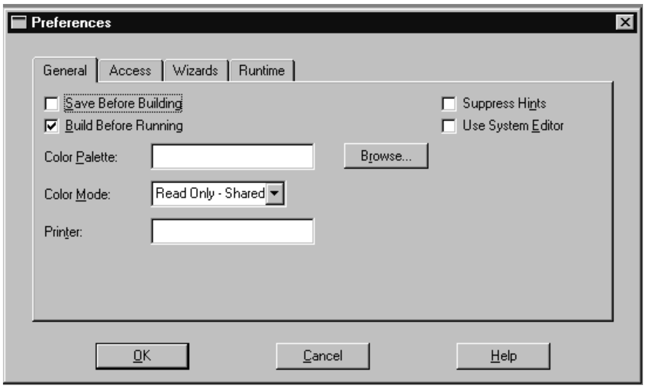
The menu example here is shown in Windows NT. However, menus appear with the same look and feel of your native GUI interface.

For example, in Motif, the Windows Print Dialog options appear as submenus of the Font menu.

Form 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	Only appears in Object Navigator; includes expand, collapse, bookmark
Program	Includes compilation and access to PL/SQL editors
Tools	Includes wizards and preferences

Customizing Your Oracle Developer Session



☐ Save Before Building
☑ Build Before Running
☐ Suppress Hints
☐ Use System Editor

Color Palette:

Color Mode: Read Only - Shared

Printer:

Copyright © Oracle Corporation, 2000. All rights reserved.

Customizing Your Oracle Forms Developer Session

What Are Oracle Forms Developer Tools Preferences?

You can use tools preferences to customize some aspects of your Oracle Forms Developer builder session.

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

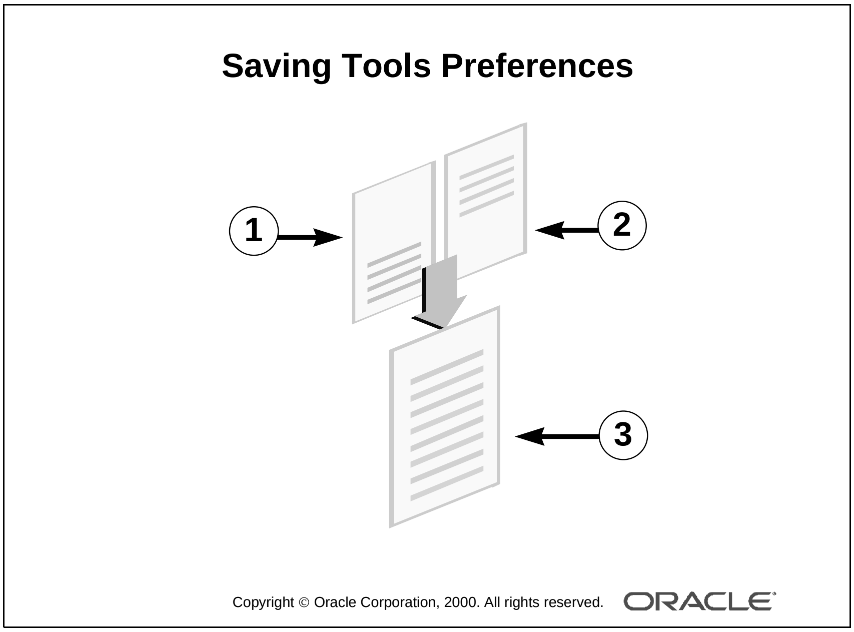
As well as session preferences, Form Builder preferences enable you to set run-time settings when running your form within the builder.

The table describes a few example form preferences.

Tab	Preference Name	Description
General	Build Before Running	Determines whether Form Builder automatically compiles the active module when you run a form. This option enables you to avoid issuing separate Compile and Run commands each time you modify and run a form.
Access	Access	Determines if form definitions are saved to (or opened from) the database or from files, or both. If access is set to both, Form Builder prompts you when you save or open each form.
Wizards	Welcome Dialog	Check box to suppress or display the first Welcome dialog box. There are several similar check boxes.
Runtime	Array Processing	Determines whether Form Builder processes groups of records at a time, reducing network traffic and increasing performance.

How to Modify Tools Preferences

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



Saving Tools Preferences

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

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

Saving Tools Preferences

When you click OK in the Preferences dialog box, Oracle Forms Developer updates your current session with the changes.

When you exit the builder, Oracle Forms Developer writes the changes to a preference file for future sessions.

Oracle Forms Developer and Oracle Reports Developer share the same preference file. If the preference file already exists, Oracle Forms Developer merges its changes with the existing file. This means that preferences for Reports are not affected.

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

Example

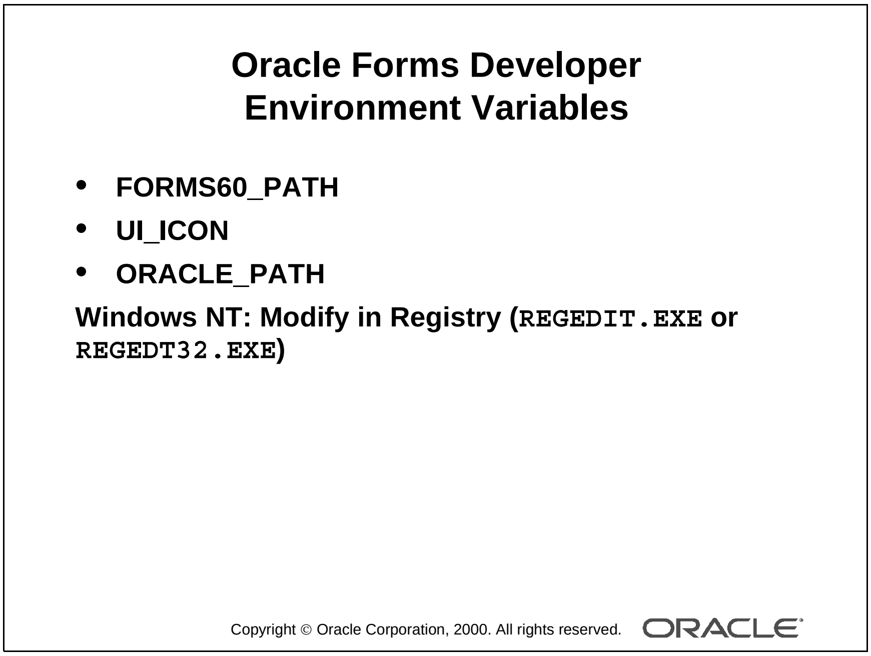
```
Reports.Object_Access = File
Forms.build_before_run = on
Forms.welcome_dialog = on
```

Oracle Forms Developer reads the preference file whenever you invoke Form Builder. Oracle Reports Developer reads the preference file whenever you invoke 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, we recommend that, where possible, you alter the options in the Preferences dialog box.



Oracle Forms Developer Environment Variables

- **FORMS60_PATH**
- **UI_ICON**
- **ORACLE_PATH**

Windows NT: Modify in Registry (`REGEDIT.EXE` or `REGEDT32.EXE`)

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Warn students to take care when modifying these variables, because they immediately affect the environment and mistakes can prevent applications from running.

Whenever possible later in the course, point out when these variables are used, and remind students to avoid hardcoding pathnames in to their applications.

Oracle Forms Developer also searches the local working directory.

Oracle Developer Environment Variables

Introduction

Oracle Forms Developer uses many environment variables. These have default values, all of which you can modify in your own environment for different applications.

In this section, we discuss only those variables that relate to file searching at run time.

Setting Pathnames

Form 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 within a form.

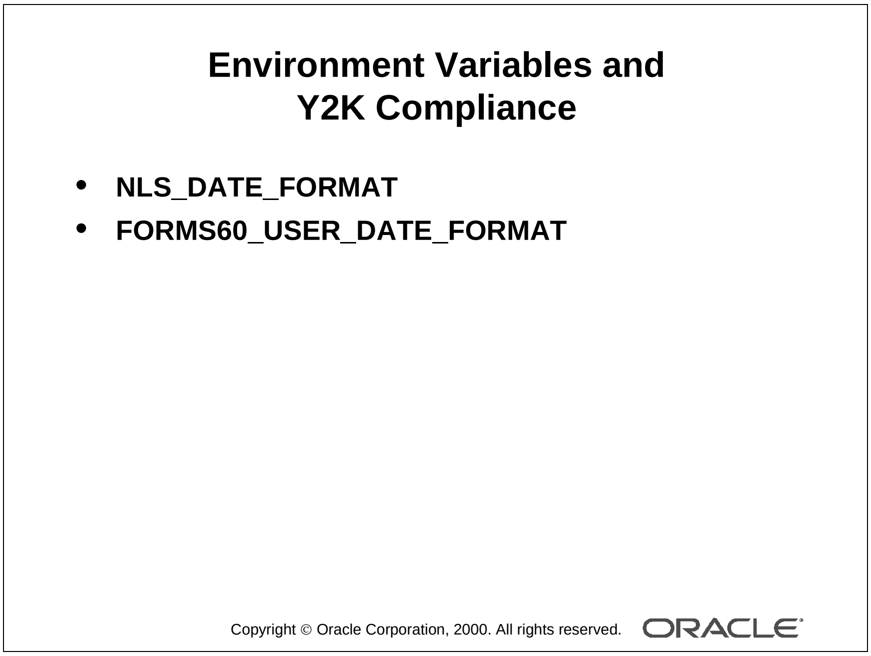
Variable	**Description**
FORMS60_PATH	A path that Form Builder searches for files at run time
UI_ICON	A path that all builders search for icon files at run time

Generic Oracle Path

ORACLE_PATH is an additional path that Form Builder searches if it cannot find a file in the specific path.

Modifying Environment Variables

In a Windows NT 32-bit environment, use the Windows Registry to modify these paths.



Environment Variables and Y2K Compliance

- **NLS_DATE_FORMAT**
- **FORMS60_USER_DATE_FORMAT**

Copyright © Oracle Corporation, 2000. All rights reserved.

Environment Variables and Y2K Compliance

Dates in Oracle Forms Developer

Dates in Oracle Forms Developer applications can come from several sources:

- Fetched from the server/database
- Entered by the end user
- Defined in the application itself

Date Format Masks

In a later lesson, you will learn how to specify a format mask for a date item in your form. In addition to the format masks a developer might explicitly specify, Form Builder uses a number of its own internal masks. The values for these internal masks can be specified with property values and environment variables.

To eliminate potential errors caused by the year 2000 (Y2K), you can set two environment variables that Form Builder uses to format date items:

- Database date format mask: Each database session within a Forms application has a single database date format mask. A default value for this mask is established by the Oracle server's initialization parameter. You can override this value in each new database session for a particular client by setting the client's NLS_DATE_FORMAT environment variable.
- Input date format mask: This mask (potentially, a set of masks) is used to convert a user-entered string into a native format date value. You can set the environment variable, FORMS60_USER_DATE_FORMAT, to specify these format masks.

Example

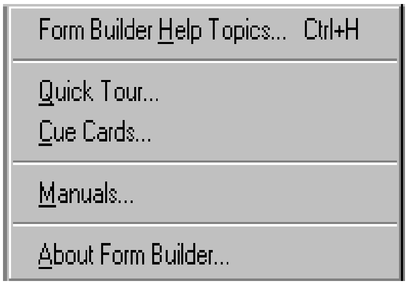
Environment Variable	**Value**
FORMS60_USER_DATE_FORMAT	`FXFMDD-MM-RRRR`

This would force the user to enter values into date items (with no specified format mask) in the format exemplified by 31-6-97. The RRRR token enables years between 1950 and 2049 to be entered with the century omitted.

Note: For more information on date handling in Oracle Forms Developer, consult the online Help in Form Builder and the *Oracle Forms Developer 6i Online Manuals*.

Using the Online Help System

- **Quick Tour: Built-in CBT package of technical information**
- **Cue Cards: Simple step-by-step instructions**



Copyright © Oracle Corporation, 2000. All rights reserved.

Invoking Online Help Facilities

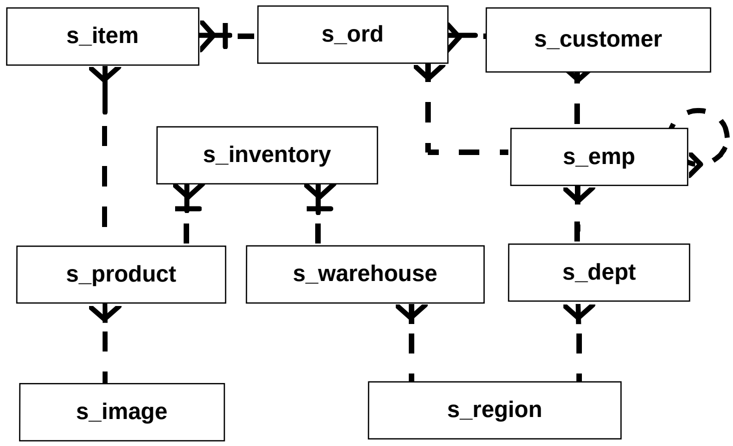
Oracle Forms Developer Help Options

The table describes the Help menu options in Form Builder.

Help Menu Option	Description
Form Builder Help Topics	This is the contents page for comprehensive online help. Includes Index and Find tabs. The Help key ([F1] (for Windows NT/95) displays context-sensitive online help at any place in the builder.
Quick Tour	This built-in computer-based training package is more detailed than the Cue Cards, including technical explanations of the Form Builder components.
Cue Cards	These appear as separate windows that give simple step-by-step instructions and examples to help novice users learn basic functionality.
Manuals	This provides an index to installed online manuals, which are HTML files that you can view with any browser.
About Form Builder	This is a separate window that shows product components and their version numbers. When you are connected to a database server, it also displays similar information for server-side product components.

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

Summit Sporting Goods Schema



s_item

s_ord

s_customer

s_inventory

s_emp

s_product

s_warehouse

s_dept

s_image

s_region

Copyright © Oracle Corporation, 2000. All rights reserved.

Introducing the Course Application

The Summit Sporting Goods Schema

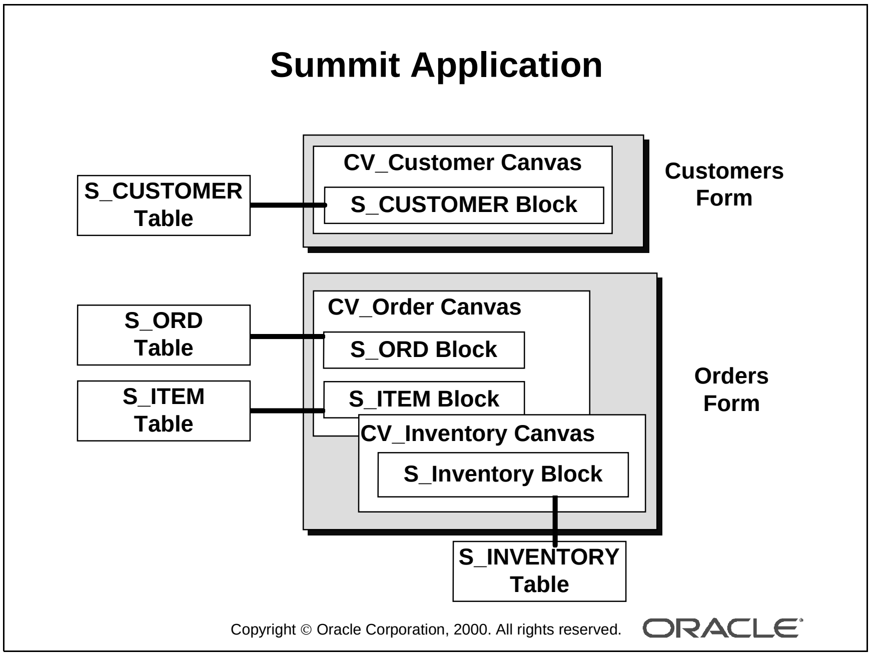
The simplified table diagram shows the tables that we use throughout this course, to build our form application. Some of you will be familiar with these tables from other Oracle courses.

Summit Sporting Goods is a company supplying 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.



Summit Application

S_CUSTOMER Table

CV_Customer Canvas

S_CUSTOMER Block

Customers Form

S_ORD Table

S_ITEM Table

CV_Order Canvas

S_ORD Block

S_ITEM Block

CV_Inventory Canvas

S_Inventory Block

Orders Form

S_INVENTORY Table

Copyright © Oracle Corporation, 2000. All rights reserved.

The Summit Sporting Goods Application

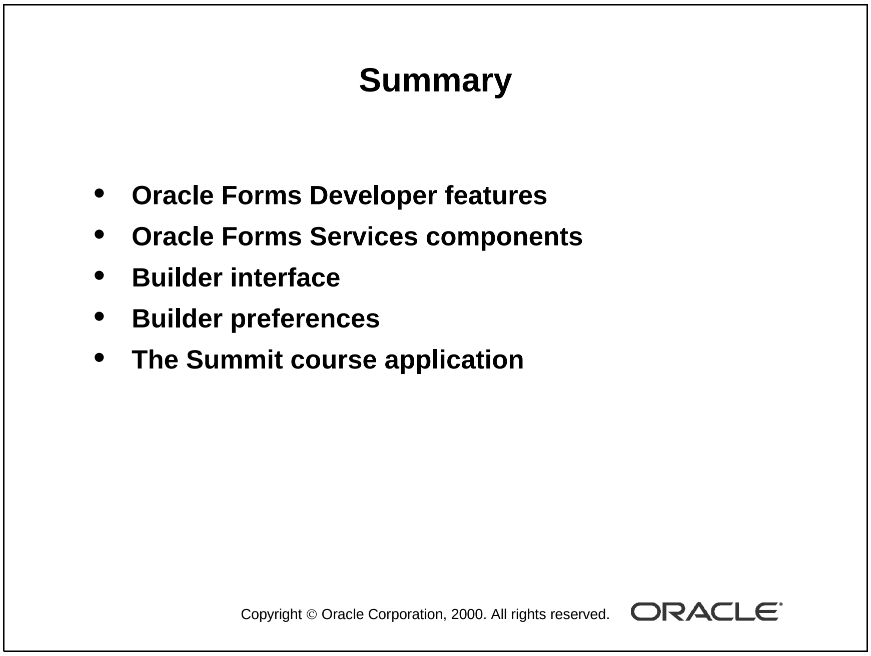
The following example Form Builder application will familiarize you with the main run-time facilities of the product. You will also build your own version of this application during the workshops in the course.

The Summit company produces a range of sporting goods, which they sell to sports stores (their customers). The Summit application is an order-entry system that maintains customer details, their orders, and the available stock (inventory).

The application consists of two main forms:

- CUSTOMERS form: Facilitates queries on existing customers and the insertion, update, or deletion of customer records. When a customer is selected, the user can open the ORDERS form to enter or view orders for that customer.

  S_CUSTOMER block: A single record block, whose base table is S_CUSTOMER.
- ORDERS form: Opened from the CUSTOMER form, the ORDERS form displays orders for a customer and the line items that belong to each order. Orders may also be created, modified, or deleted in this form. You can also display the stock available on the ordered products.
  - S_ORD block: A single record master block for the form (The base table is S_ORD, but the block also displays associated information from other tables, such as the name of the customer.)
  - S_ITEM block: The related detail block for an order, showing its line items and the products ordered (This is a multirecord block whose items are on the same canvas as those in the S_ORD block. The ITEM block’s base table is S_ITEM, but it also displays information from other tables, such as the product name and standard price.)
  - S_INVENTORY block: A multirecord block showing warehouse stock for a product (Its items are on a separate canvas, which is assigned to its own window. This block is linked to the current product in the S_ITEM block, but the two blocks can operate independently.)



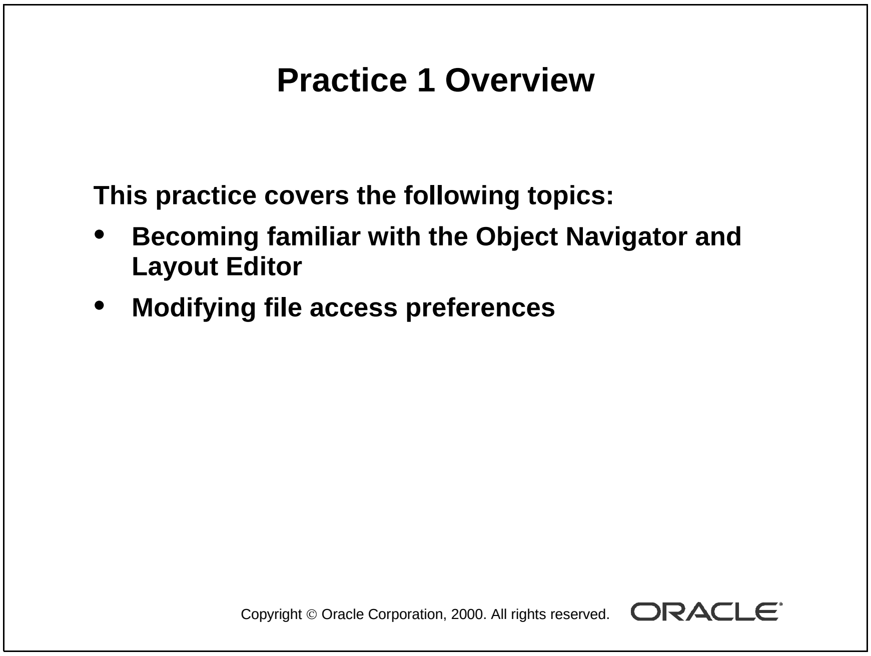
Summary

- **Oracle Forms Developer features**
- **Oracle Forms Services components**
- **Builder interface**
- **Builder preferences**
- **The Summit course application**

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

- The Oracle Forms Developer provides a builder interface, including Object Navigator, Layout Editor, PL/SQL Editor, and Property Palette components, and offers a comprehensive online Help system.
- Oracle Forms Developer has a set of preferences that you can alter for the current and subsequent builder sessions.
- Finally, in this lesson we described the Summit application tables that we use throughout the course to build our forms.



Practice 1 Overview

This practice covers the following topics:

- **Becoming familiar with the Object Navigator and Layout Editor**
- **Modifying file access preferences**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 1 in Appendix A, “Practice Solutions.”

Practice 1 Overview

This practice familiarizes you with Oracle Forms Developer by asking you a number of questions that you answer by navigating around the Form Builder interface.

- Becoming familiar with the Layout Editor and Object Navigator in Form Builder
- Modifying file access preferences

Instructor Note

This practice is intended to make students familiar and at ease with the interface. For additional help with the Layout Editor, refer them to Appendix F, Using the Layout Editor.

Practice 1

1 Invoke Form Builder, and select “Open an existing form” from the Welcome page.

2 Open the `Orders.fmb` form module from the Open Dialog window.

3 Change your preferences so that when you open or save a file, Form Builder gives you the option of saving the file to the file system or the database.

4 Try to open the `Customers.fmb` form module. Notice that the module access dialog box is displayed. Click Cancel.
Modify your preferences so that Form Builder will access the file system only.

5 Close the ORDERS form.

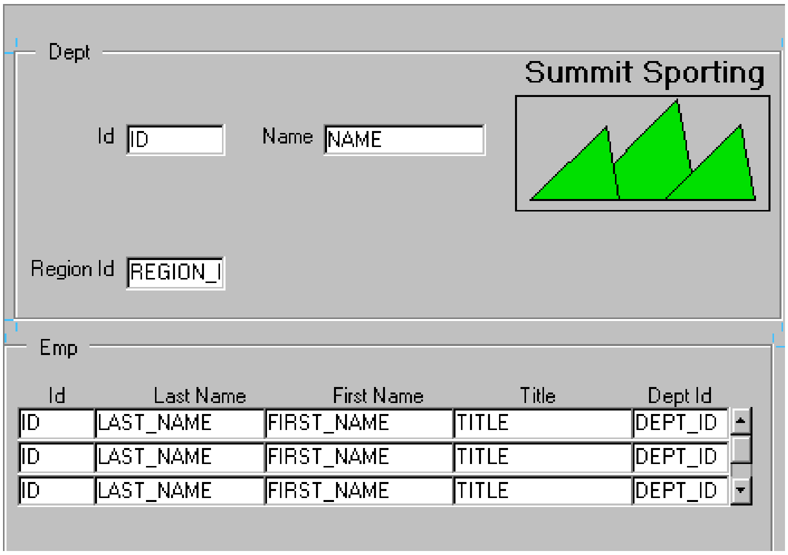
6 Open the `Summit.fmb` form module.

7 Expand the Data Blocks node.

8 Expand the Database Objects node. If you cannot expand the node, connect to the database and try again. What do you see below this node?

9 Collapse the Data Blocks node.

10 Change the layout of the `Summit.fmb` form module to match the screen shot shown below. At the end, save your changes, and exit Form Builder.



Id	Last Name	First Name	Title	Dept Id
ID	LAST_NAME	FIRST_NAME	TITLE	DEPT_ID
ID	LAST_NAME	FIRST_NAME	TITLE	DEPT_ID
ID	LAST_NAME	FIRST_NAME	TITLE	DEPT_ID

a Invoke the Layout Editor.

b Move the three summit shapes to the top-right corner of the layout. Align the objects along the bottom edge.

c Select the summit shape in the middle and place it behind the other two shapes.

d Draw a box with no fill around the summit shapes.

e Add the text Summit Sporting on top of the box.

f Move the Name, Id, and Region_Id items to match the screenshot.

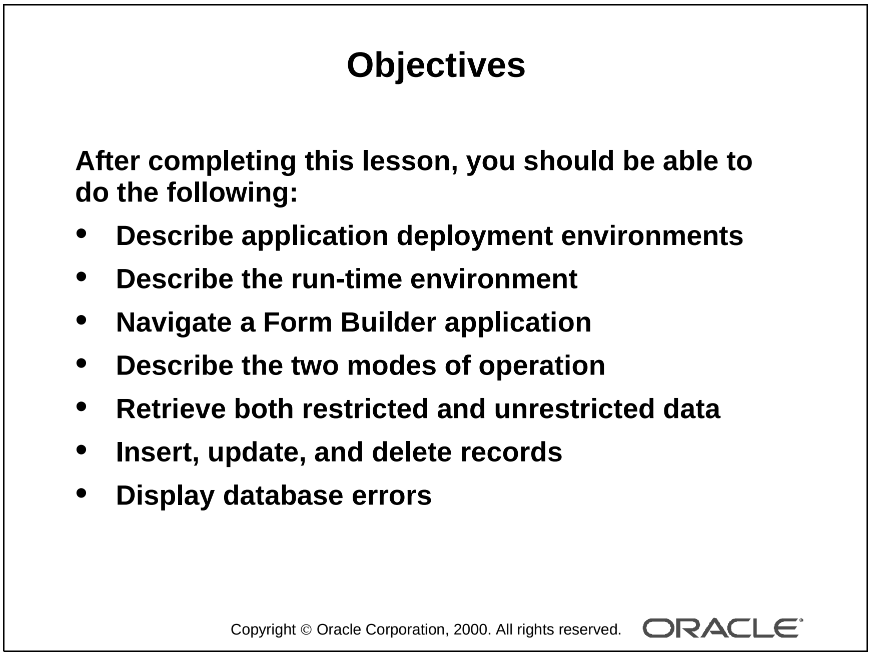
g Move the First_Name item up to align it at the same level as the Last_Name item.

h Resize the scroll bar, to make it the same height as the three records in the Emp block.

i Save the form module, and exit Form Builder.

2

Running a Form Builder Application



Objectives

After completing this lesson, you should be able to do the following:

- **Describe application deployment environments**
- **Describe the run-time environment**
- **Navigate a Form Builder application**
- **Describe the two modes of operation**
- **Retrieve both restricted and unrestricted data**
- **Insert, update, and delete records**
- **Display database errors**

Copyright © Oracle Corporation, 2000. All rights reserved.

Schedule

Topic	**Timing**
Lecture	50 minutes
Practice	25 minutes
Total	75 minutes

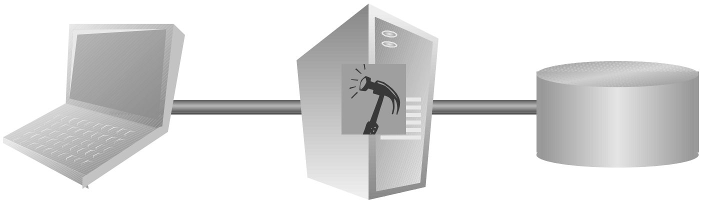
Introduction

Overview

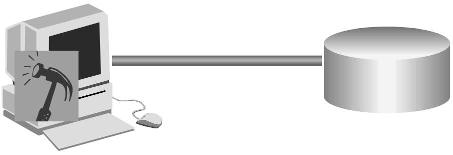
In this lesson, you will run an existing application both on the Web and in a client-server environment in order to become familiar with the run-time interface of the Oracle Forms Developer. You must understand the form operator environment before designing and building your own applications.

Application Deployment

- **Oracle Forms Services Web deployment**

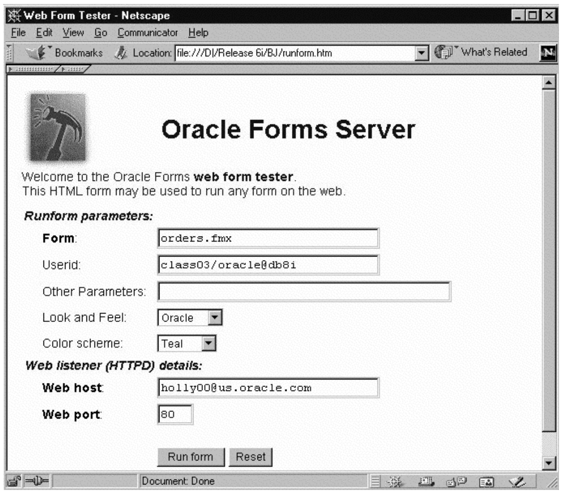


- **Oracle Forms Developer client-server environment**



Copyright © Oracle Corporation, 2000. All rights reserved.

Web Form Tester


Web Form Tester - Netscape
File Edit View Go Communicator Help
Bookmarks
What's Related
Run form
Reset
Document: Done

Location: file:///D|/Release 6i/BJ/runform.htm

Oracle Forms Server

Welcome to the Oracle Forms **web form tester**.
This HTML form may be used to run any form on the web.

Runform parameters:

Form: orders.fmx

Userid: class03/oracle@db8i

Other Parameters:

Look and Feel: Oracle

Color scheme: Teal

Web listener (HTTPD) details:

Web host: holly00@us.oracle.com

Web port: 80

Copyright © Oracle Corporation, 2000. All rights reserved.

Application Deployment

Form Builder applications can be deployed to end users in a Web implementation or a client-server implementation.

Web Implementation

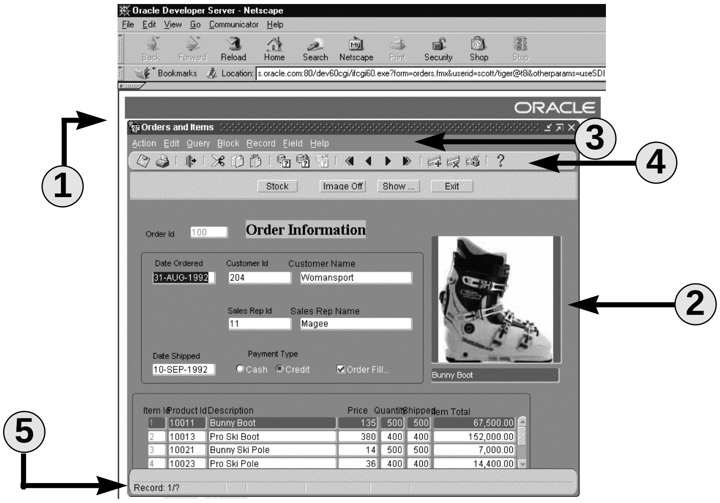
Deploying form applications to the Web is implemented by the three-tier architecture of Oracle9*i* Application Server. Application logic and the Forms Services run-time engine reside on the middle-tier application server rather than on desktop client machines. All trigger processing occurs on database and application servers, while user interface processing occurs on the Forms client.

In this course, you will use the Web form tester to deploy your form applications to the Web.

Client-Server Implementation

In this two-tier architecture, application logic and the Forms Services run-time engine reside on end users' desktop machines. Although application logic can include database server-side triggers and logic, typically all user interface and trigger processing occurs on client machines.

Web Deployment: Run-Time Components


Oracle Developer Server - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location:
ORACLE
Orders and Items
Action Edit Query Block Record Field Help
Stock
Image Off
Show ...
Exit
Order Information
Bunny Boot
1
2
3
4
5

Location: s.oracle.com:80/dev60cgi/ifcgi60.exe?form=orders.fmx&userid=scott/tiger@t8i&otherparams=useSDI

Order Id 100

Date Ordered 31-AUG-1992 | Customer Id 204 | Customer Name Womansport

Sales Rep Id 11 | Sales Rep Name Magee

Date Shipped 10-SEP-1992 | Payment Type ○ Cash ○ Credit | ☑ Order Fill...

Item Id	Product Id	Description	Price	Quantity	Shipped	Item Total
1	10011	Bunny Boot	135	500	500	67,500.00
2	10013	Pro Ski Boot	380	400	400	152,000.00
3	10021	Bunny Ski Pole	14	500	500	7,000.00
4	10023	Pro Ski Pole	36	400	400	14,400.00

Record: 1/?

Copyright © Oracle Corporation, 2000. All rights reserved.

1	Browser window
2	Java applet
3	Default Menu
4	Menu toolbar
5	Console

Instructor Note

Demonstration Launch Netscape Navigator. Run the Orders form on the Web by double-clicking the shortcut on your desktop for Run a Form on the Web. Enter `orders.fmx` for the Form parameter and enter a username, password, and connect string for the Userid parameter. Select the Run form button. Point out the Browser window, Java applet, Default menu, Menu toolbar, and Console components. Terminate this run-time session by exiting the form, and then closing the browser window. Be sure to advise students to always exit the form to terminate the applet before closing the browser window.

Run the Orders form again using the Web form tester. This time, select `Oracle` for the Look and Feel parameter and `Teal` for the Color scheme parameter. Point out the rounded appearance of the buttons and the new look of the Menu toolbar.

Web Deployment: What You See at Run Time

Starting a Run-Time Session

A *run-time session* consists of one or more linked form and menu modules, all under the control of a single user.

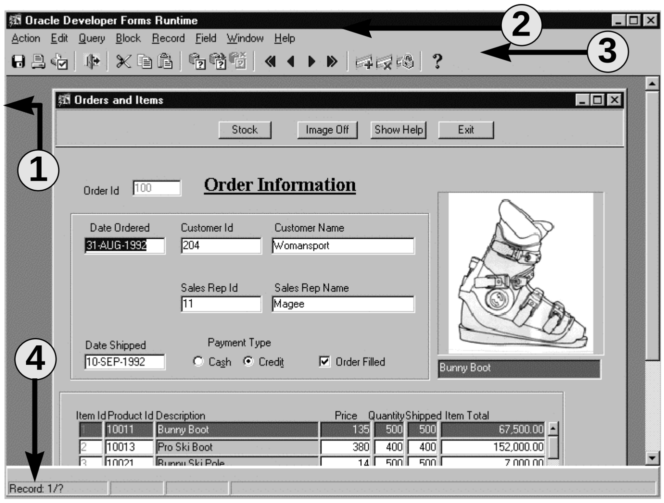
To start a Forms run-time session on the Web, you simply have to point your browser to a uniform resource locator (URL) that corresponds to a Hypertext Markup Language (HTML) file. The HTML file contains all necessary tags, parameters, and parameter values required to run the selected Forms application on the Web. The following sequence of events occurs:

1. The user accesses the URL of an HTML file that indicates that a Forms application should be run.
2. The Oracle HTTP Server receives an HTTP request from the browser client and contacts the Forms CGI.
3. The Forms CGI dynamically creates an HTML page containing all the information to start the Forms session.
4. The Oracle HTTP Server downloads a generic applet to the client. The client caches the applet so that the applet does not need to be downloaded again. The applet will run all future Forms applications for as long as it resides in the client's cache.
5. The client applet contacts the Forms Listener to start the session. The Forms Listener starts an instance of the Forms Runtime Engine on the Forms Server. If included in the HTML file, Forms Runtime command-line parameters (such as form name, user ID and password, database SID, and so on) and any user-defined Form Builder parameters are passed to the process by the Forms Listener.
6. The Listener establishes a connection with the Runtime Engine and sends the connection information to the Forms applet.
7. The Forms applet then establishes a direct connection with the Runtime Engine. The Forms applet and Runtime Engine then communicate directly, freeing the Listener to accept startup requests from other users.
8. The Forms applet displays the application's user interface in the main window of the user's Web browser.

Technical Note

More information on Oracle Forms Server is covered in the course *Oracle 9iAS Forms Services: Deploy Internet Applications*. Also, see *Deploying Forms Applications to the Web with Oracle Internet Application Server.*

Client-Server Run-Time Components


Oracle Developer Forms Runtime
Action Edit Query Block Record Field Window Help
Orders and Items
Stock
Image Off
Show Help
Exit
Order Information
1
2
3
4

Order Id 100

Date Ordered 31-AUG-1992

Customer Id 204

Customer Name Womansport

Sales Rep Id 11

Sales Rep Name Magee

Date Shipped 10-SEP-1992

Payment Type ○ Cash ● Credit

☑ Order Filled

Bunny Boot

Item Id	Product Id	Description	Price	Quantity	Shipped	Item Total
1	10011	Bunny Boot	135	500	500	67,500.00
2	10013	Pro Ski Boot	380	400	400	152,000.00
3	10021	Bunny Ski Pole	14	500	500	7,000.00

Record: 1/?

Copyright © Oracle Corporation, 2000. All rights reserved.

1	MDI parent window
2	Default menu
3	Menu toolbar
4	Console

Client-Server Environment: What You See at Run Time

Starting a Run-Time Session

In a GUI environment, you usually start a Forms run-time session by selecting one of the following items:

- An icon in the window system
- A menu option in the window system

In either case, this executes a run-time command to start a session. (The command name varies according to the platform).

For example:

```
IFRUN60 my_form scott/tiger
```

In a GUI environment, the Forms Run time Options screen is displayed when you execute the command (IFRUN60 in Microsoft Windows). Use this screen to enter the form name, the database user ID, and other options, if you did not specify them in the initial command.

What Is the MDI Parent Window?

The *multiple-document interface (MDI) parent window* is the container window in which your application can display multiple form modules (document windows). The title of the MDI parent window is Oracle Developer Forms Runtime.

Note: The concept of MDI is only applicable to Microsoft Windows.

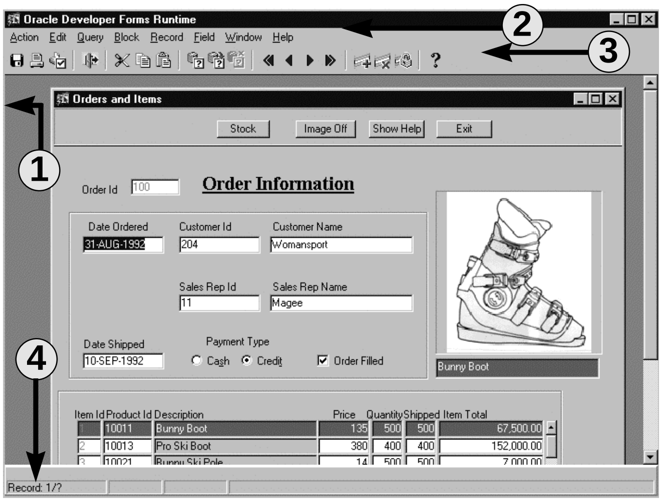
What Is the SDI Window?

Although MDI is the default system of window management during run time, Form Builder also provides support for a *single-document interface* (SDI) *window* on Microsoft Windows.

What Is the Default Menu?

The *Default menu,* which is part of all Oracle Forms Developer applications, is an alternative to keystroke operations. You can replace or customize the Default menu to introduce your own functionality into a form module.

Client-Server Run-Time Components


Oracle Developer Forms Runtime
Action Edit Query Block Record Field Window Help
?
Orders and Items
Stock
Image Off
Show Help
Exit
1
2
3
4

Order Id 100

Order Information

Date Ordered 31-AUG-1992
Customer Id 204
Customer Name Womansport
Sales Rep Id 11
Sales Rep Name Magee
Date Shipped 10-SEP-1992
Payment Type ○ Cash ● Credit
☑ Order Filled

Bunny Boot

Item Id	Product Id	Description	Price	Quantity	Shipped	Item Total
1	10011	Bunny Boot	135	500	500	67,500.00
2	10013	Pro Ski Boot	380	400	400	152,000.00
3	10021	Bunny Ski Pole	14	500	500	7,000.00

Record: 1/?

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Launch the Forms Runtime component. Run the Orders form by using `Orders.fmx`. Point out the MDI parent window, Default menu, Menu toolbar, and Console components.

What Is the Menu Toolbar?

The *Menu toolbar* contains buttons corresponding to menu items. At run time, it appears above any user-defined toolbars. It executes the same code as menu items, and it is a shortcut to menu commands that does not duplicate code or effort.

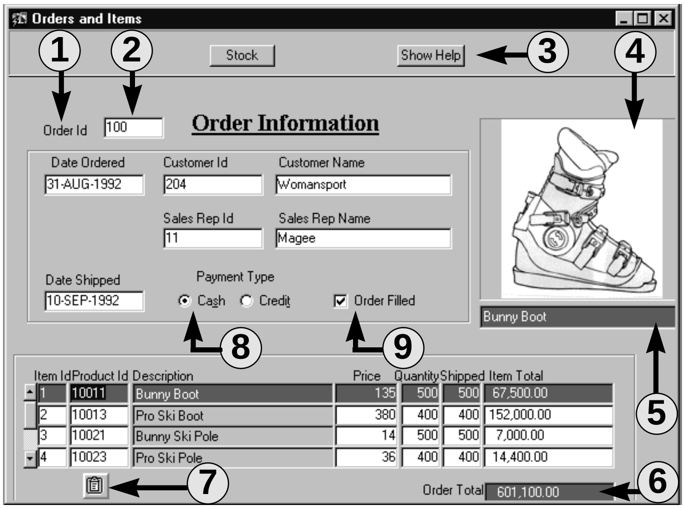
What Is the Console?

The *console* is the generic name for the standard features that provide information at run time. The console is displayed at the bottom of the window and consists of:

- The message line that displays both Form Builder and application-specific messages.
- The status line that displays a variety of indicators to reflect the current state of the form module.

Record: *n/m*	The *n*th record retrieved and displayed so far, out of *m* number of total records that can be retrieved by the query. Until the last record is viewed, *m* displays a “?”; after that it displays the number corresponding to the last record.
Enter-Query	The current block is in Enter Query mode and no records have been retrieved.
List of Values	A list of values (LOV) is associated with the current item.

Data Elements



Order Id 100

Order Information

Date Ordered	Customer Id	Customer Name
31-AUG-1992	204	Womansport

Sales Rep Id	Sales Rep Name
11	Magee

Date Shipped 10-SEP-1992

Payment Type ● Cash ○ Credit

☑ Order Filled

Bunny Boot

Item Id	Product Id	Description	Price	Quantity	Shipped	Item Total
1	10011	Bunny Boot	135	500	500	67,500.00
2	10013	Pro Ski Boot	380	400	400	152,000.00
3	10021	Bunny Ski Pole	14	500	500	7,000.00
4	10023	Pro Ski Pole	36	400	400	14,400.00

Order Total 601,100.00

Copyright © Oracle Corporation, 2000. All rights reserved.

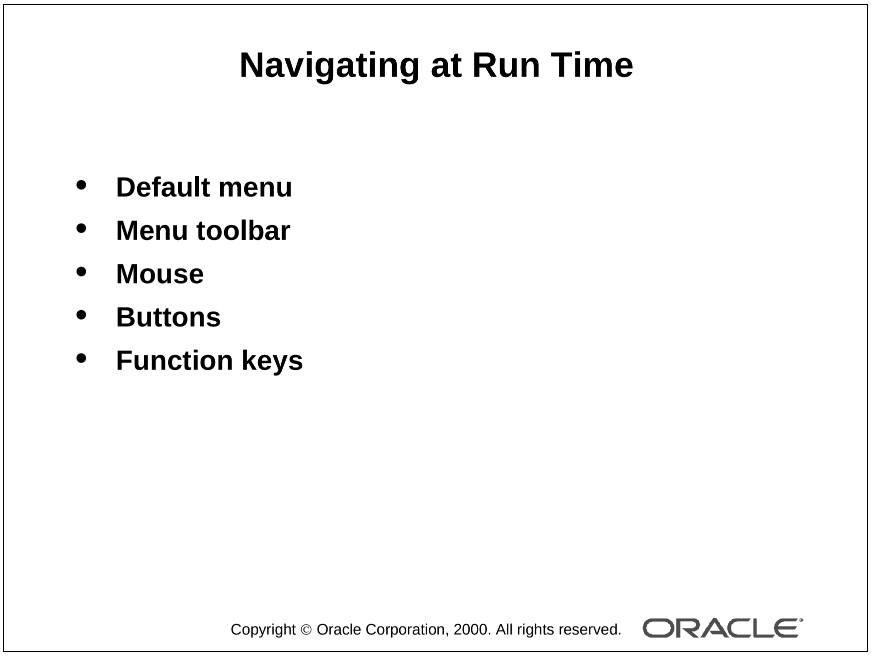
1	Prompt
2	Text item
3	Button
4	Image item
5	Display item
6	Calculated field
7	Iconic button
8	Radio group
9	Check box

Identifying the Data Elements

Data elements in a Form Builder application are the same, regardless of whether the application is deployed on the Web or in a client-server environment:

- Boilerplate text
- Boilerplate graphics
- Prompts
- Text items
- List items
- Push buttons
- Display items
- Radio buttons
- Check boxes
- Image items
- Sound items
- Hierarchical Tree items
- Chart items
- Custom items

The environment determines the appearance (layout and size of objects do not change) of the application and the data elements. The visually appealing characteristics that display when you run the application in a bitmapped environment are limited when you run it in a character mode environment.



Navigating at Run Time

- Default menu
- Menu toolbar
- Mouse
- Buttons
- Function keys

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration In the ORDERS form, point out the main data elements. Show the different means of navigation.

Navigating a Form Builder Application

The Default Menu

The Default menu is automatically available in a form, unless it is disabled or replaced with a customized menu. Select options from the menu by using the mouse or function keys. At run time, use the menu to perform the following tasks:

- Move the cursor and navigate between data blocks, records, and items.
- Save or clear all changes.
- Execute queries.
- Insert new records or delete existing records.
- Invoke Help.

The Menu Toolbar

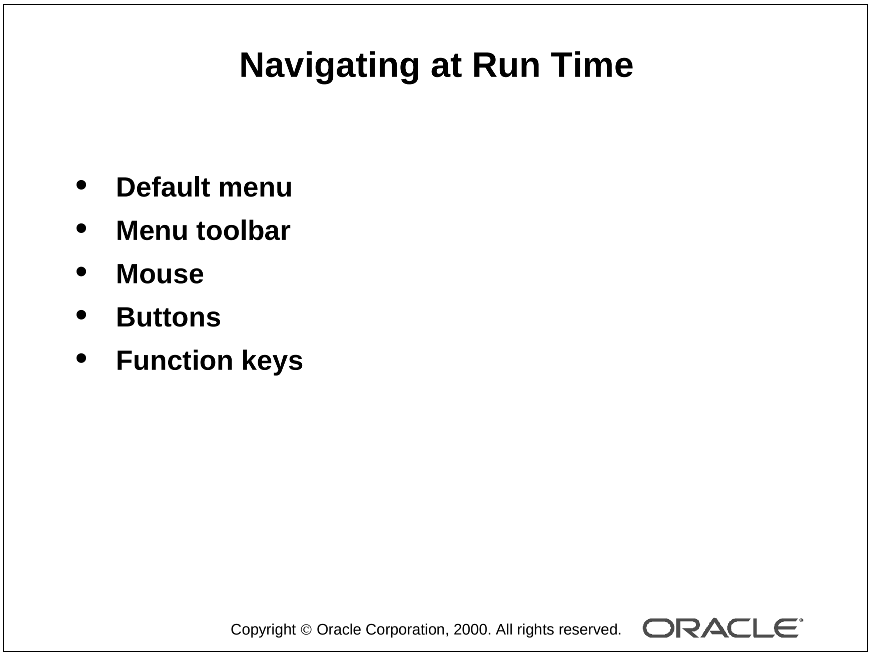
You can use the Default menu toolbar buttons to perform the following operations also available through the Default menu:

- Save all changes.
- Exit the form.
- Execute queries.
- Navigate between data blocks or records.
- Insert new records or delete existing records.
- Invoke Help to see properties of an item.

The Mouse

You can use the mouse to navigate and to perform many user operations in a bitmapped environment without needing to learn the function keys. Use the mouse to perform the following actions:

- Move the cursor.
- Select from a menu.
- Select from an LOV.
- Select or clear a check box.
- Select a button, including a radio group button.
- Switch to an open window.
- Respond to an alert.
- Scroll records or lines by using a data block or item scroll bar.
- Manipluate a custom item.



Navigating at Run Time

- **Default menu**
- **Menu toolbar**
- **Mouse**
- **Buttons**
- **Function keys**

Copyright © Oracle Corporation, 2000. All rights reserved.

Buttons

Web applications often use buttons as a means of navigation. You can click buttons with the mouse. Use buttons to perform the following tasks:

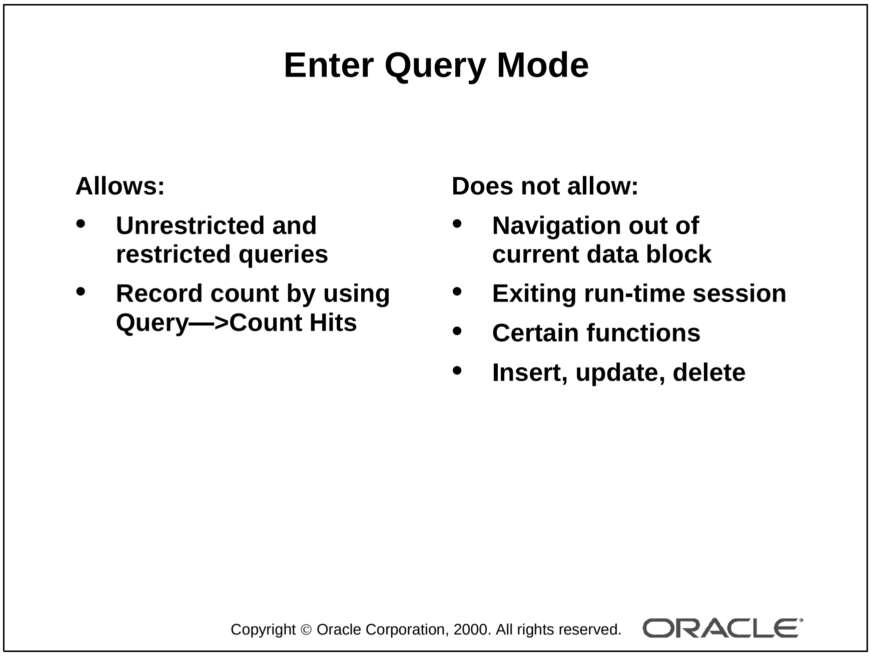
- Move input focus.
- Display a LOV.
- Invoke an editor.
- Invoke another window.
- Commit data.
- Issue a query.
- Perform calculations.
- Exit the form.

Function Keys

In addition to navigating with the mouse, you can move from item to item in sequence with function keys. Use function keys to perform the following tasks:

- Navigate between data blocks, records, and items.
- Execute queries.
- Insert new records or delete existing ones.
- Invoke Help.

To view a list of keys and the functions they perform, select [Ctrl]+K.



Enter Query Mode

Allows:

- **Unrestricted and restricted queries**
- **Record count by using Query—>Count Hits**

Does not allow:

- **Navigation out of current data block**
- **Exiting run-time session**
- **Certain functions**
- **Insert, update, delete**

Copyright © Oracle Corporation, 2000. All rights reserved.

Modes of Operation

Form Builder has two main modes of operation: Enter Query mode and Normal mode.

Enter Query Mode

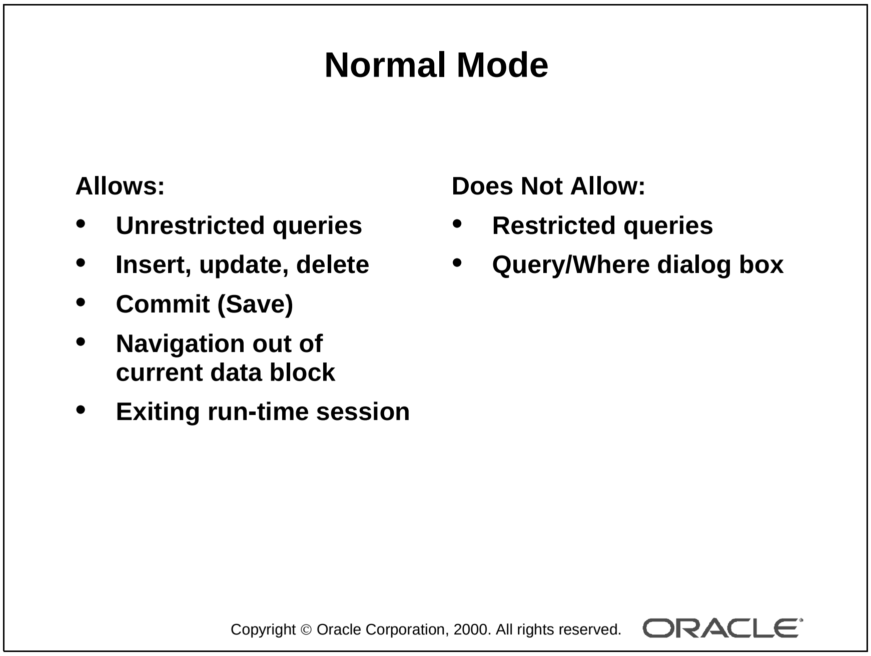
Use *Enter Query mode* to enter search criteria for a database query. In Enter Query mode, your keystrokes are interpreted as search criteria for retrieving restricted data.

What You Can Do in Enter Query Mode

- Retrieve all records.
- Retrieve records by using selection criteria.
- Retrieve records by using the Query/Where dialog box.
- Obtain the number of records that will be retrieved before fetching them from the database by using Query—>Count Hits.

What You Cannot Do in Enter Query Mode

- Navigate out of the current block.
- Exit from the run-time session.
- Use certain functions, such as Next Record.
- Insert new records.
- Update existing records.
- Delete records.



Normal Mode

Allows:

- **Unrestricted queries**
- **Insert, update, delete**
- **Commit (Save)**
- **Navigation out of current data block**
- **Exiting run-time session**

Does Not Allow:

- **Restricted queries**
- **Query/Where dialog box**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Normal Mode

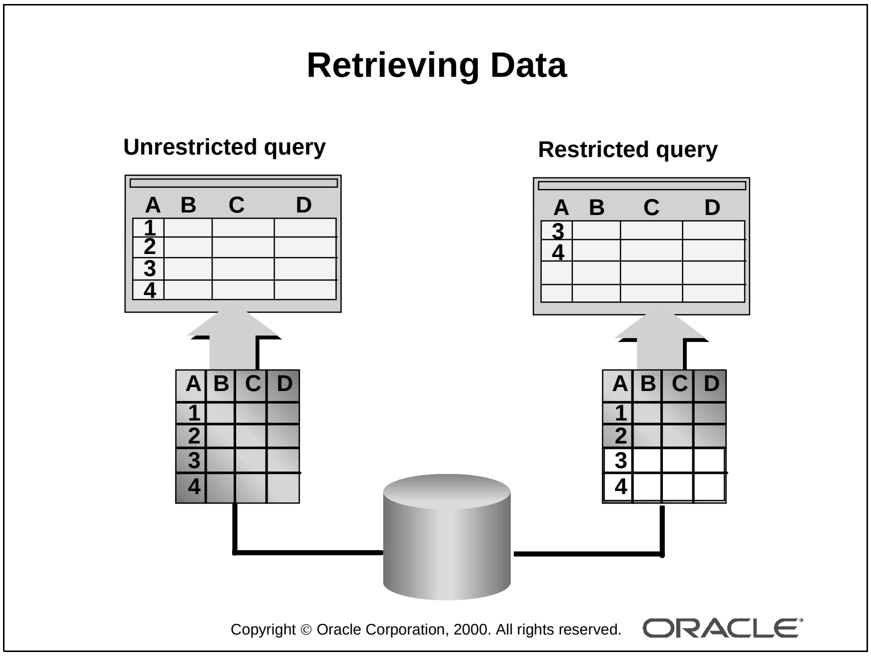
Use *Normal mode* to insert and alter records in the database. In Normal mode, your keystrokes are interpreted as either the entering of new records or the altering of existing ones.

What You Can Do in Normal Mode

- Retrieve all records.
- Insert new records.
- Update records.
- Delete records.
- Commit (Save) records.
- Rollback (Clear) records.
- Navigate outside of the current data block.
- Exit the run-time session.

What You Cannot Do in Normal Mode

- Retrieve a restricted set of records.
- Invoke the Query/Where dialog box.



Retrieving Data

Unrestricted query

Restricted query

Copyright © Oracle Corporation, 2000. All rights reserved.

Retrieving Data

You can use a form module to retrieve information from the database without knowing any SQL syntax. However, if you are an experienced SQL user, you may want to supplement Oracle Forms Developer default processing with your own SQL predicates. There are two general types of queries:

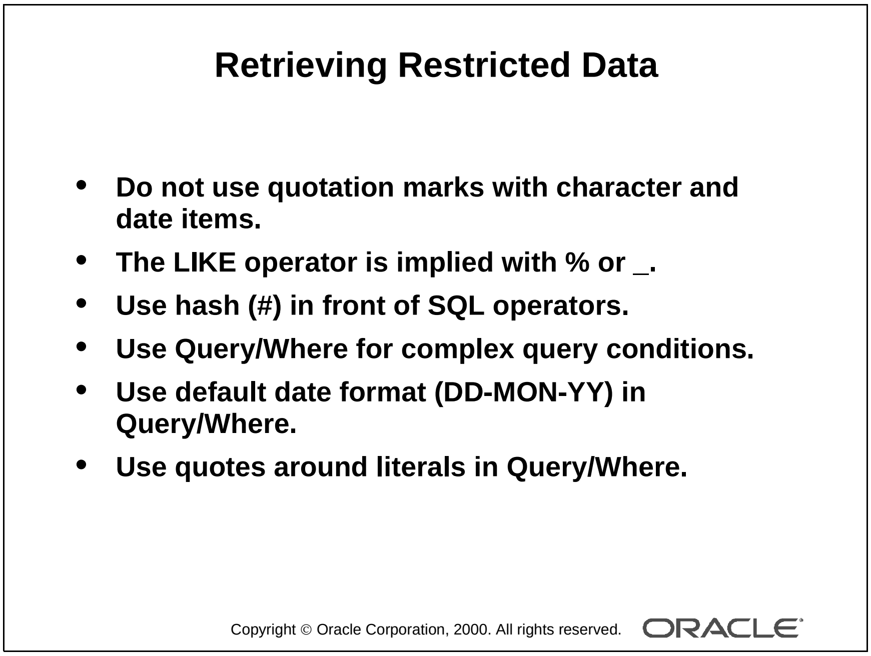
Query Type	**Description**
Unrestricted (global query)	The equivalent of selecting all the rows for all the represented columns from the base table for the queried data block
Restricted	The equivalent of selecting a restricted set of rows for all the represented columns from the base table for the queried data block

Performing an Unrestricted Query

You can retrieve unrestricted data by performing one of the following actions:

- Select Query—>Execute.
- Press the appropriate function key.
- Click the Execute Query button.

Note: You cannot perform a query while you have unsaved updates, inserts, or deletes. Either save or undo the changes before you continue with the query.



Retrieving Restricted Data

- **Do not use quotation marks with character and date items.**
- **The LIKE operator is implied with % or _.**
- **Use hash (#) in front of SQL operators.**
- **Use Query/Where for complex query conditions.**
- **Use default date format (DD-MON-YY) in Query/Where.**
- **Use quotes around literals in Query/Where.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The default date format is dependent on the language/territory. In Oracle8*i* Release 2 (8.1.6), the default date formats were modified, replacing ’YY’ with ’RR’ and ’YYYY’ with ’RRRR’. For example, if NLS_LANG is set to Finland, then the default date format will be DD.MM.RRRR.

Performing a Restricted Query

You can perform a restricted query by using any one of the following methods:

- Matching values
- Matching patterns (wildcards)
- A Query/Where dialog box for user entry of SQL predicates

Valid Search Criteria

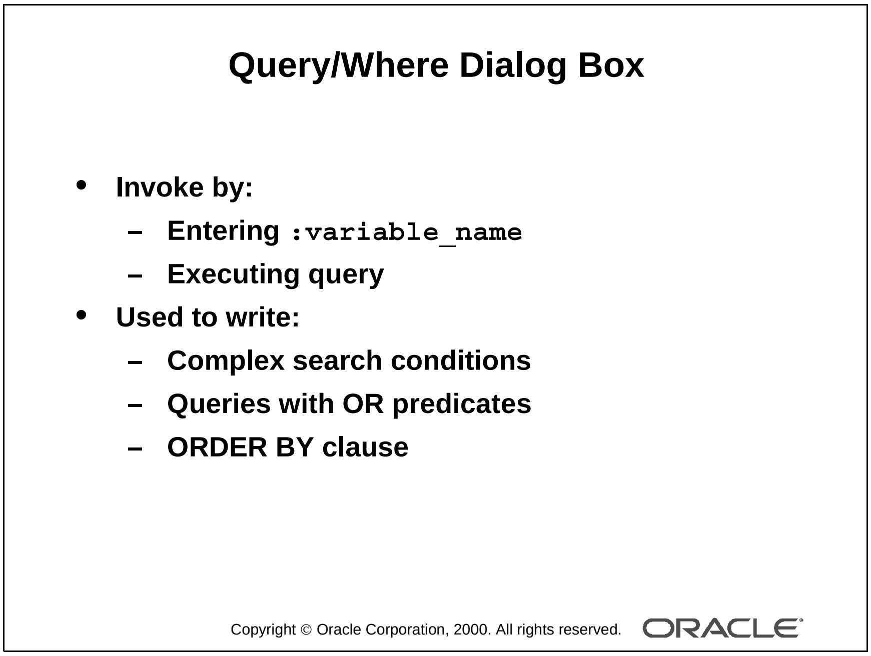
Item	**Criterion**	**Uses**
Order ID	110	Exact match
Customer ID	%6	Implied LIKE operator
Order ID	#BETWEEN 105 AND 107	BETWEEN operator
Date Ordered	09/09/1992	Exact match by comparing the date, using the default format (09-SEP-92)
Payment Type	CASH	Exact match
Sales Rep ID	:S	Query Where dialog

How to Perform a Restricted Query

1. Do one of the following:
   - Select Query—>Enter.
   - Click the Enter Query button.
   - Press the appropriate function key.

   Enter-Query displays on the status line.
2. Enter search criteria into appropriate items.
3. Do one of the following:
   - Select Query—>Execute.
   - Click the Execute Query button.
   - Press the appropriate function key.

Note: Form Builder constructs a select statement by using the AND operator for all specified conditions.



Query/Where Dialog Box

- **Invoke by:**
  - **Entering `:variable_name`**
  - **Executing query**
- **Used to write:**
  - **Complex search conditions**
  - **Queries with OR predicates**
  - **ORDER BY clause**

Copyright © Oracle Corporation, 2000. All rights reserved.

Using the Query/Where Dialog Box

The *Query Where dialog box* is a window in which you enter complex search criteria by using raw SQL. Using the Query/Where dialog effectively requires knowledge of SQL. Use Query/Where to perform the following tasks:

- Write complex search conditions.
- Write queries with OR predicates.
- Order the result of a query.

Note: Form Builder logically uses the AND operator to append the Query/Where conditions to any other search criteria (including those imposed by the form designer) and constructs a SELECT statement.

Example

To restrict the query to orders with a Sales Rep ID (:S) of 11 OR an Order ID (:O) between 100 and 200, enter the following in the Query/Where dialog box:

```
:S = 11 OR :O between 100 and 200
```

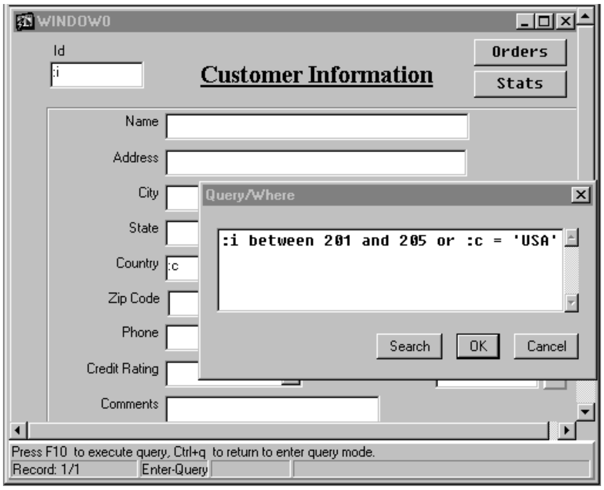
Example

To sort the data by Sales rep ID (:S), enter the following in the Query/Where dialog box:

```
ORDER BY :S
```

If you enter an ORDER BY at run time, it overrides any ordering defined by the designer.

Query/Where Dialog Box



Id :i

Name

Address

City

State

Country :c

Zip Code

Phone

Credit Rating

Comments

```
:i between 201 and 205 or :c = 'USA'
```

Press F10 to execute query, Ctrl+q to return to enter query mode.

Record: 1/1 Enter-Query

Copyright © Oracle Corporation, 2000. All rights reserved.

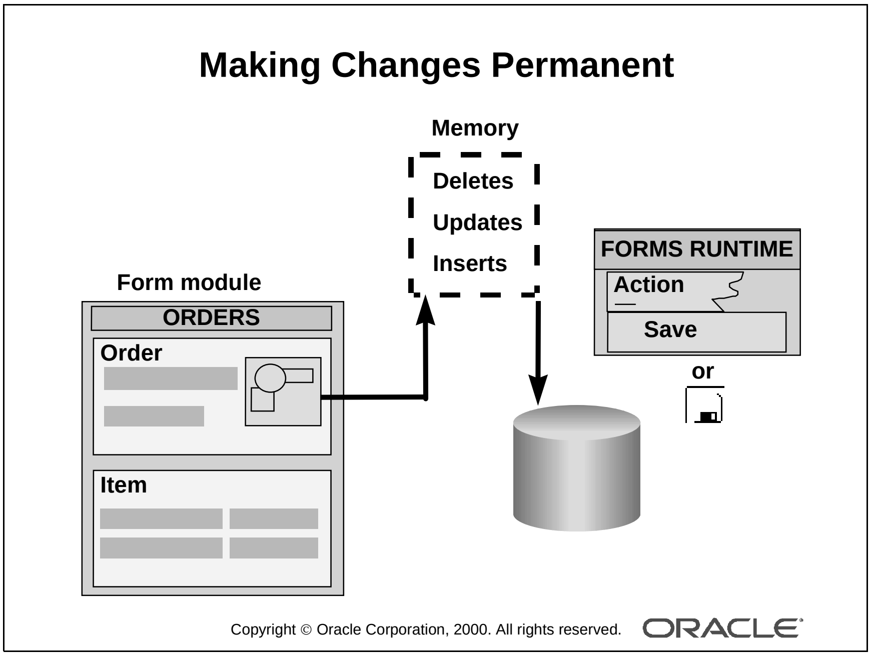
How to Use the Query/Where Dialog Box

1. Do one of the following:
   - Select Query—>Enter.
   - Click Enter Query.
   - Press the appropriate function key.
2. Enter a colon (:) followed by a unique character variable name in one or more items.
3. Do one of the following:
   - Select Query—>Execute.
   - Click Execute Query.
   - Press the appropriate function key.

   Note: Alternatively, you can select Query—>Count Hits if you simply want to know how many records will match your criteria.

   The Query/Where dialog box is displayed.
4. Enter the search criteria by using variables, SQL, and logical operators.
5. Click OK.

Note: To perform a query without any variables, type only the colon (:) and execute the query. Doing so also displays the Query/Where dialog box.



Making Changes Permanent

Memory

Deletes

Updates

Inserts

FORMS RUNTIME

Action

Save

or

Form module

ORDERS

Order

Item

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Inserting, Updating, and Deleting Records

Upon entering a typical form module you are in Normal mode. This means that Form Builder regards anything you type into a blank record as an insert and anything you type over an existing record as an update.

How to Insert a Record

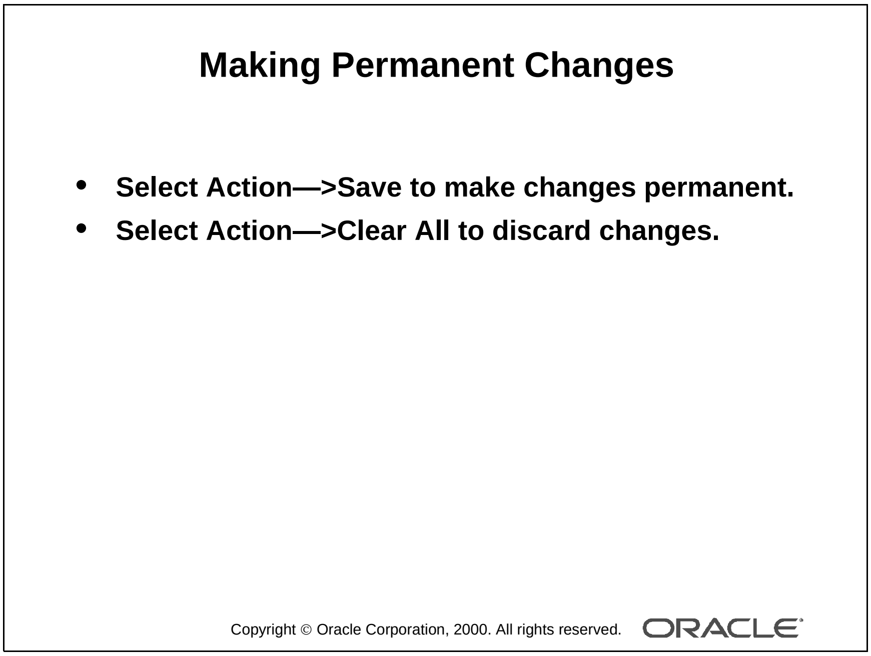
1. Ensure that you have the cursor positioned on a blank record by taking one of the following actions:
   - Scroll down until you find one (always the last in the block).
   - Select Record—>Insert.
   - Click Insert Record (green +).
   - Press the appropriate function key.
2. Enter the data into the relevant items.

How to Update a Record

1. Select Query—>Enter.
2. Enter the search criteria to retrieve the appropriate record.
3. Select Query—>Execute to retrieve all records that satisfy your specific search criteria.
4. Scroll through the records, stopping at the record to be updated.
5. Update the record.

How to Delete a Record

1. Select Query—>Enter.
2. Enter the search criteria to retrieve the appropriate record.
3. Select Query—>Execute to retrieve all records that satisfy your specific search criteria.
4. Scroll through the records, stopping at the record to be deleted. Delete the record by taking one of the following actions:
   - Select Record—>Remove to clear the record and mark it for deletion.
   - Click Remove Record (red X) to clear the record and mark it for deletion.
   - Press the appropriate function keys.



Making Permanent Changes

- **Select Action—>Save to make changes permanent.**
- **Select Action—>Clear All to discard changes.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Making Inserts, Updates, and Deletes Permanent

To make any inserts, updates, or deletes permanent you must save (commit) them to the database. To do this, take one of the following actions:

- Select Action—>Save.
- Click Save in the menu toolbar.

Discarding Inserts, Updates, and Deletes

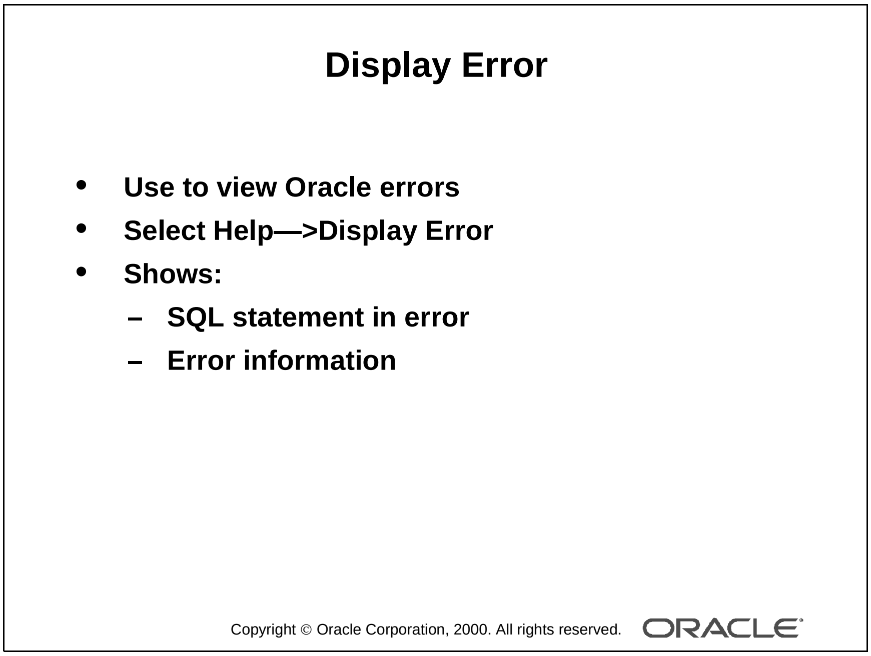
To discard any inserts, updates, or deletes, you must clear the records (rollback) instead of saving. Perform a rollback by selecting Action—>Clear All.

Exiting a Run-Time Session

In a GUI environment, you exit the run-time session by taking one of the following actions:

- Select Action—>Exit.
- Click Exit.
- Press the appropriate function keys.

Note: By default, you cannot exit the form while you have unsaved updates, inserts, or deletes. You need to either save or undo the changes before you can exit.



Display Error

- **Use to view Oracle errors**
- **Select Help—>Display Error**
- **Shows:**
  - **SQL statement in error**
  - **Error information**

Copyright © Oracle Corporation, 2000. All rights reserved.

Displaying Errors

If an Oracle error is displayed on the message line while you are operating a form application, you can view the underlying SQL code by selecting Help—>Display Error.

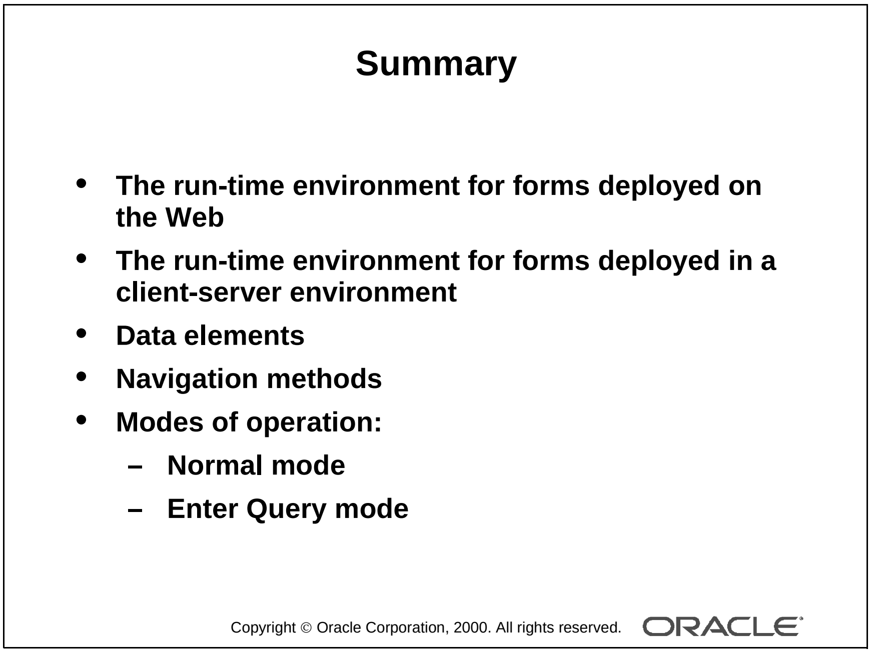
Example

Here is the SQL statement in error and its corresponding error:

```
SELECT deptno,dname,loc,ROWID
FROM dept
WHERE (deptno in (‘a’,’b’))

ORA-01722: invalid number
```

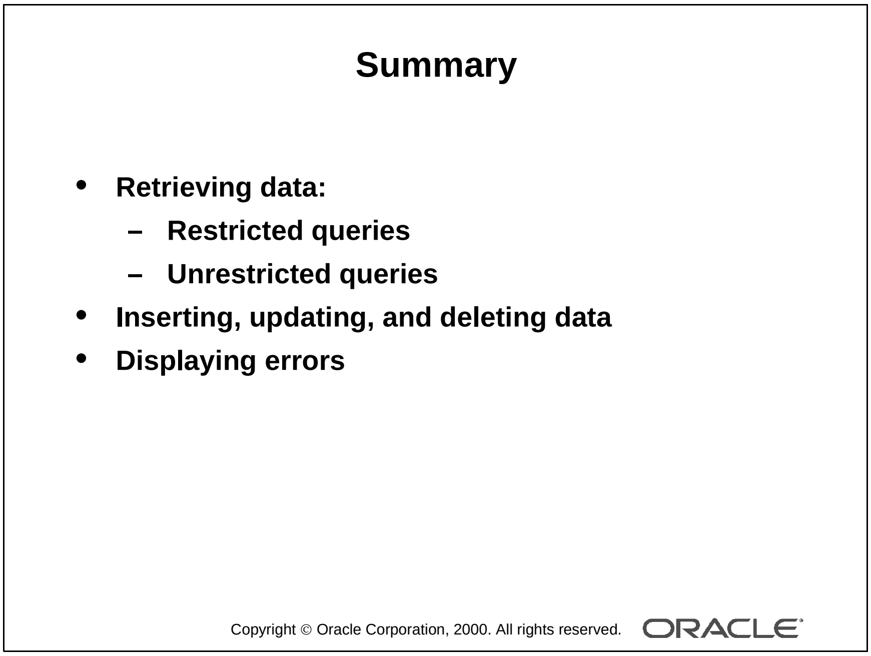
Note: Selecting Help—>Display Error displays only those errors where the error on the message line is preceded by `ORACLE error`.



Summary

- The run-time environment for forms deployed on the Web
- The run-time environment for forms deployed in a client-server environment
- Data elements
- Navigation methods
- Modes of operation:
  - Normal mode
  - Enter Query mode

Copyright © Oracle Corporation, 2000. All rights reserved.



Summary

- Retrieving data:
  - Restricted queries
  - Unrestricted queries
- Inserting, updating, and deleting data
- Displaying errors

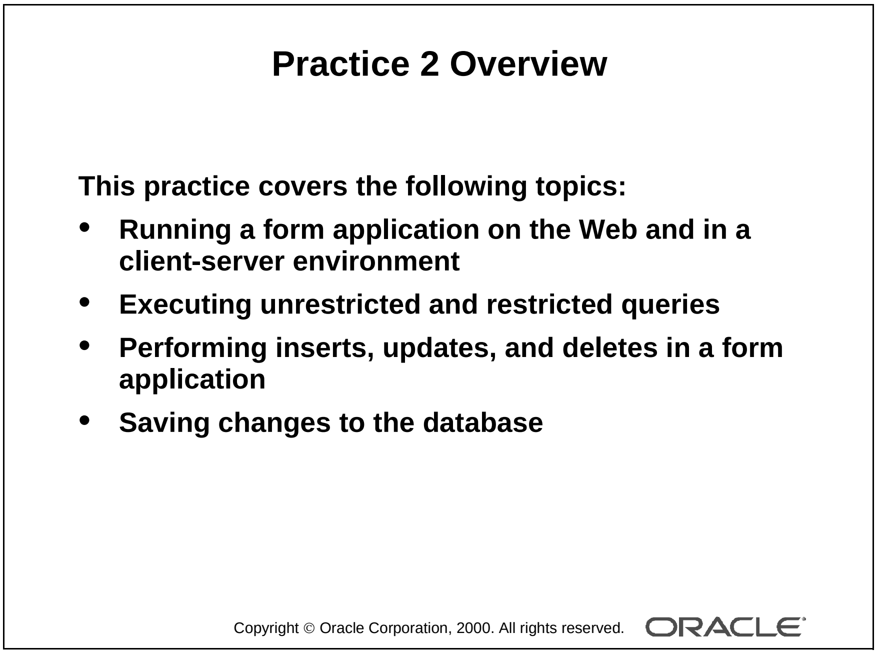
Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

This lesson introduced the operator interface of Form Builder. The following concepts were covered in this lesson:

- The run-time environment for forms deployed on the Web
- The run-time environment for forms deployed in a client server environment
- The data elements of a form
- Navigation methods
- Modes of operation:
  - Normal mode
  - Enter Query mode
- Retrieving data by performing:
  - Restricted queries—you supply search criteria
  - Unrestricted queries—you supply no search criteria
- Inserting, updating, and deleting records
- Displaying error information, when an error is reported by Form Builder

Many other facilities are available when operating a form, depending on what the form designer has implemented. These other Form Builder facilities are discussed in the coming lessons.



Practice 2 Overview

This practice covers the following topics:

- **Running a form application on the Web and in a client-server environment**
- **Executing unrestricted and restricted queries**
- **Performing inserts, updates, and deletes in a form application**
- **Saving changes to the database**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 2 in Appendix A, “Practice Solutions.”

Practice 2 Overview

In this practice session, you use the Summit application to query the database tables. You also insert, update, and delete data through the application.

- Running a form application
- Executing unrestricted queries to query database tables
- Executing restricted queries and specify search criteria
- Performing insert, update, and delete operations in a form application
- Saving changes to the database

Practice 2

Running a Forms Application on the Web

1 Run the form module Customers on the Web.

Double-click the shortcut on your desktop for Run a Form on the Web. Enter the Runform parameters and run the Customers form.

2 Select Help—>Keys from the menu.

3 Execute an unrestricted query.

4 Execute a restricted query to retrieve information about the customer with the ID of 212.

5 Exit this run-time session.

6 Start another run-time session on the Web using the form module Customers. For this session, set the Runform parameter Look and Feel to `Oracle`. Notice the different appearance of the user interface.

7 Execute a restricted query to retrieve the “Womansport” record.
Notice that the status line displays the words `ENTER QUERY`.
Notice that only the one record is retrieved.

8 Execute a restricted query to retrieve customers with a Sales Rep ID greater than 13.

9 Try each of these restricted queries:

- Retrieve all cities starting with `San`.
- Retrieve all those customers based in the USA with a credit rating of Excellent.

10 Display the customer details for Big John’s Sports Emporium and click the Orders button to move to the Orders form module.

11 Click the Image Off button and notice that the image item no longer displays. Click the Image On button and notice that the image item displays.

12 Query only those orders with a payment type of Credit.

13 Move to the first record in the Item block and click the Stock button. The Inventory block displays in a separate window. Execute a query to get stock information.

14 Exit the run-time session.

Running a Forms Session in Client-Server

1 Start a Forms Runtime session by using the form module Customers.

2 Select Help—>Keys from the menu. Notice that the function keys are defined differently in a client-server environment than they are for Web deployment.

3 For the customer Big John’s Sports Emporium, insert a new record in the ORDER block, as detailed below.
Notice that some items are already populated with default values. Enter the following:

Item	**Value**
Date Shipped	Today’s date (*DD-MON-YYYY*)
Payment Type	Cash (Radio group button)
Order Filled	No (Unchecked)

4 Insert a new record in the ITEM block.
Move to the ITEM block and enter the following:

Item	**Value**
Product ID	`50530`
Quantity	`2`

5 Save the new records.

6 Update the order that you have just placed and save the change.

7 Attempt to delete the order that you have just placed. What happens?

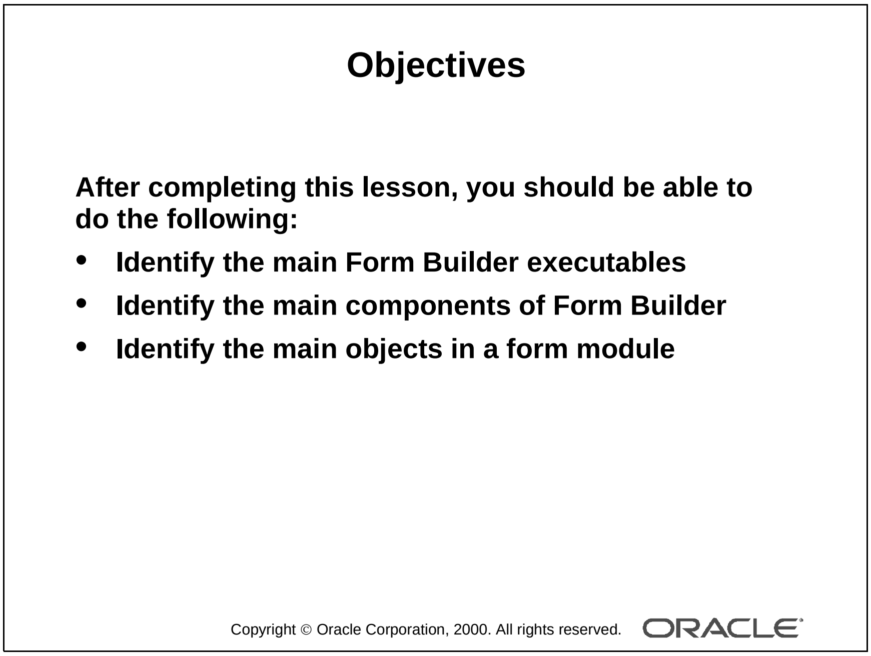
8 Delete the line item for your order and save the change.

9 Now attempt to delete your order and save the change.

10 Exit the run-time session.

3

Working in the Form Builder Environment



Objectives

After completing this lesson, you should be able to do the following:

- Identify the main Form Builder executables
- Identify the main components of Form Builder
- Identify the main objects in a form module

Copyright © Oracle Corporation, 2000. All rights reserved.

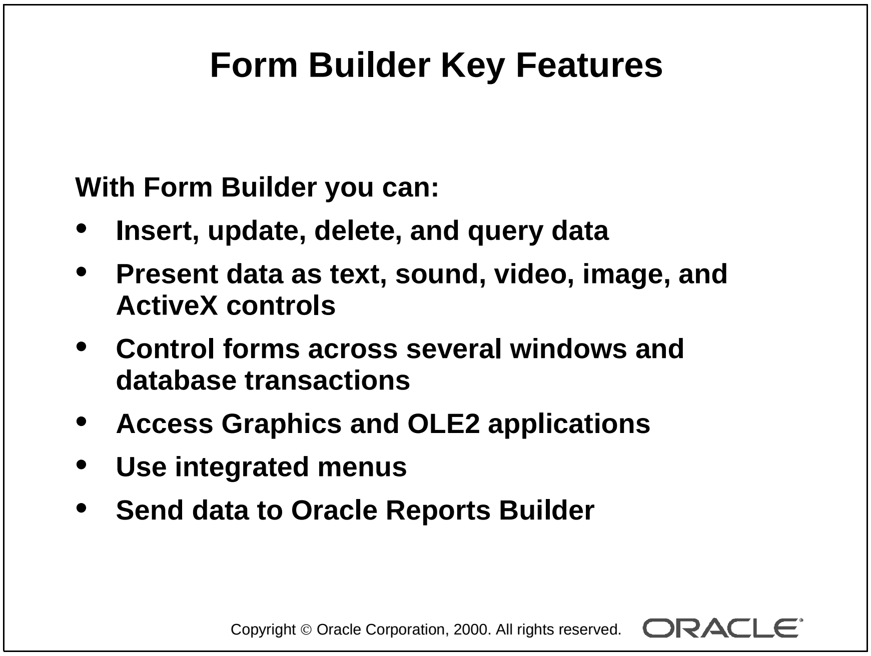
Schedule

Topic	**Timing**
Lecture	45 minutes
Total	45 minutes

Introduction

Overview

This lesson provides you with an overview of Form Builder, including a high-level description of its components and object hierarchy. Using this knowledge, you can plan and implement the structure of your form applications.



Form Builder Key Features

With Form Builder you can:

- **Insert, update, delete, and query data**
- **Present data as text, sound, video, image, and ActiveX controls**
- **Control forms across several windows and database transactions**
- **Access Graphics and OLE2 applications**
- **Use integrated menus**
- **Send data to Oracle Reports Builder**

Copyright © Oracle Corporation, 2000. All rights reserved.

What Is Form Builder?

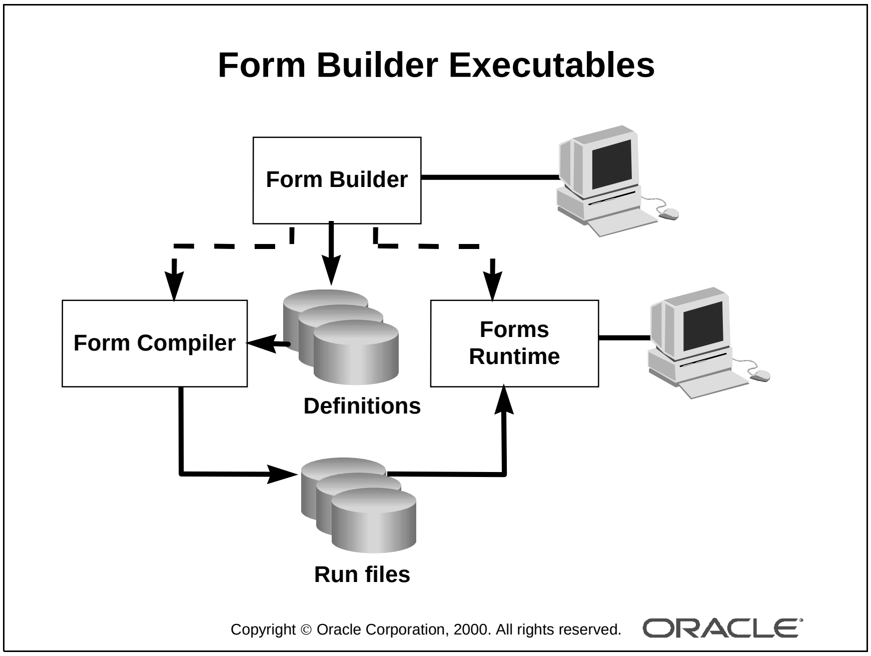
Form Builder is a major component of Oracle Forms Developer. It enables you to quickly develop form-based applications for presenting and manipulating data in a variety of ways.

With Form Builder applications your users can:

- Insert, update, delete, and query data by using a variety of interface items
- Present data using text, sound, video, image, and custom controls, including JavaBeans and ActiveX
- Control forms across several windows and database transactions
- Access the facilities of Oracle Graphics Builder and OLE2 applications directly
- Access comprehensive facilities by using integrated menus
- Send data directly to Oracle Reports Builder

As the designer of Form Builder applications, you can:

- Design forms that use a number of data sources, including Oracle databases
- Build applications quickly and easily by using powerful GUI development tools
- Design applications for Internet deployment
- Design applications that are portable across both GUI and character-mode environments
- Copy and move objects and their properties easily between applications
- Use design features such as wizards, the Layout Editor, Object Navigator, and PL/SQL Editor



Form Builder Executables

Form Builder

Form Compiler

Definitions

Forms Runtime

Run files

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Instructor Note

Explain the components of the product and their relationships.

Form Builder Executables

Form Builder includes three executables (components) that you can access as the designer of applications.

Form Builder

This is the application-building component of Oracle Forms Developer. This component lets you design and store the definitions of form, menu, and library documents. While in the Form Builder, you can invoke the other two components, Form Compiler and Forms Runtime. You must run the Form Builder component in a GUI environment in order to use its graphical design facilities.

Form Compiler

Once your form is built, use the Form Compiler. This reads the definition of your module and creates an executable run file.

Forms Runtime

This is the program that runs an executable form application, and the modules within it. The files used at run time must already have been compiled by the Form Compiler component.

Invoking Form Builder Executables

In a GUI environment, you usually store commands to invoke Form Builder components in menus and window icons for convenient access. You can also enter these commands on the command line.

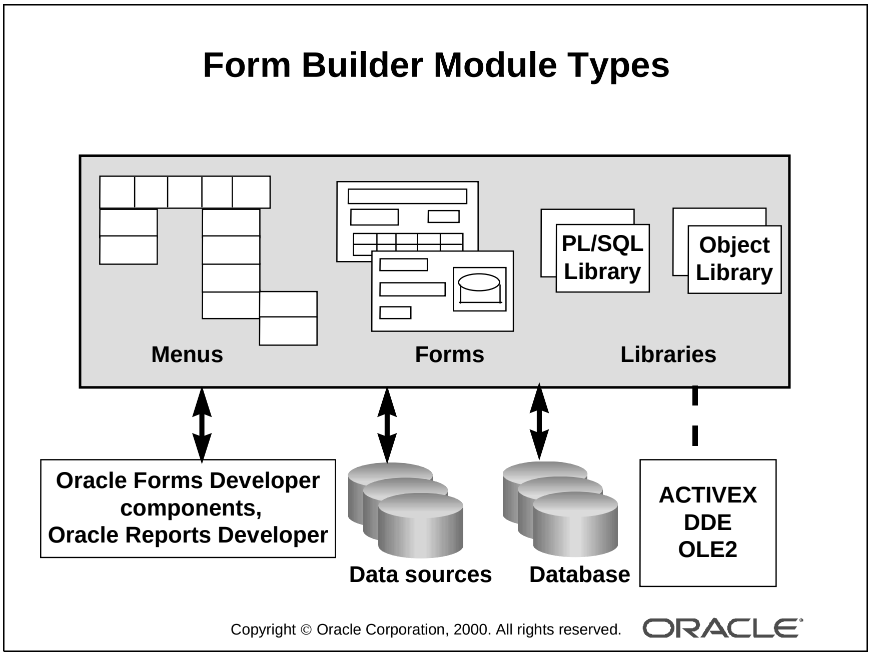
For example:

```
IFRUN60 my_form scott/tiger
```

Note: Commands for invoking the product components vary according to platform.

Instructor Note

If students want to know how to invoke other Forms Builder components from the command line, refer them to online Help. Enter `Form Builder components` as the search string and then select `starting` as the subtopic.



Form Builder Module Types

Menus

Forms

Libraries

PL/SQL Library

Object Library

Oracle Forms Developer components,
Oracle Reports Developer

Data sources

Database

ACTIVEX
DDE
OLE2

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Explain the relationship among multiple forms, menus, and libraries.

Form Builder Module Types

A Form Builder application can consist of many modules—that is, files. A module is a major component of your application and is the basis for storage and ownership. A module owns the rest of the objects in the system.

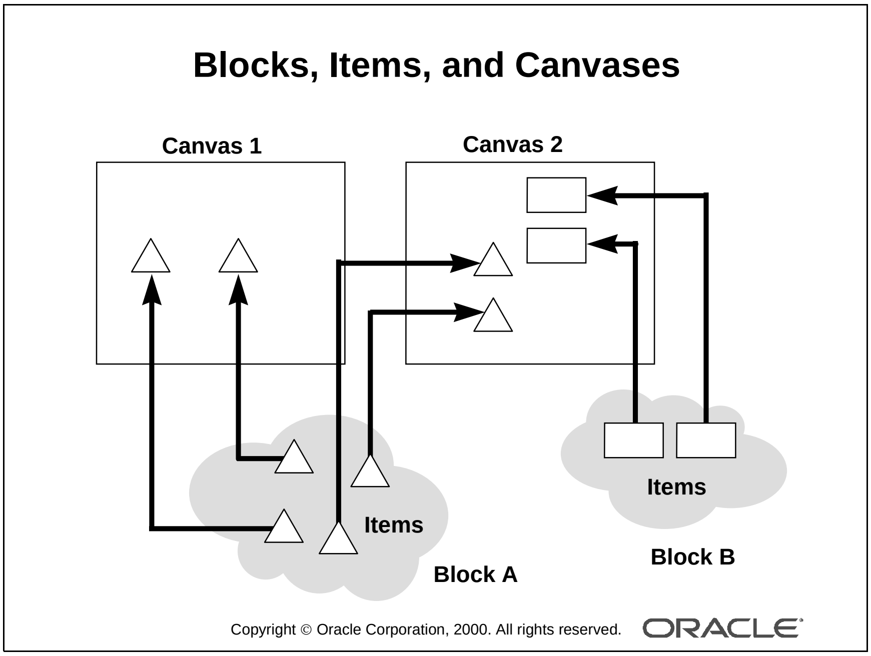
A Form Builder module can be of the following types:

- Form: As the main component of an application, the form module presents the objects and data that users can see or interact with. Data items in a form are arranged into records.
- Menu: A menu module can consist of a hierarchy of menus, each with selectable items.
- PL/SQL Library: A PL/SQL Library is a collection of PL/SQL program units whose code can be referenced and called from other modules.
- Object Library: An Object Library is a collection of form objects that you can use in other modules. You can create it to store, maintain, and distribute standard objects that can be reused across the entire development organization.

Form Builder provides the default menu for every form. The default menu includes commands for all basic database operations, such as insert, update, delete, query, and so on. If your application has specific requirements that are not met by the default menu, you can create a custom menu module. Menu modules are usually attached to form modules. In this way, the menus can provide a service to the facilities offered by a form, as well as options to invoke facilities elsewhere.

PL/SQL Library documents can contain program units that can be used by other form and menu modules.

You can build an application from multiple form modules, menu modules, and library documents as needed.



Blocks, Items, and Canvases

Copyright © Oracle Corporation, 2000. All rights reserved.

Form Builder Components

Form modules make up the main “body” of a Oracle Forms Developer application. They can consist of many object types, and some of these objects are visible to the user at run time.

The three major objects in a form are:

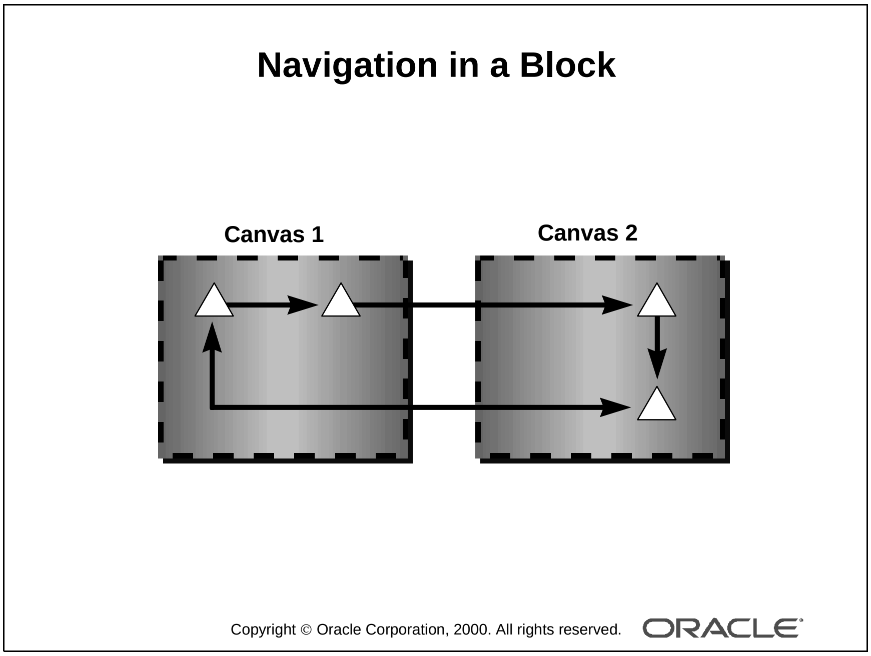
- Blocks: A block is the intermediate building unit for forms. Each form consists of one or more blocks. A block is the logical owner of items, rather like the sections of a paper form. Each item in a form belongs to a block. Items in one block are logically related; for example, they may correspond to columns in the same database table or may need to be part of the same navigation cycle.

  Blocks therefore provide a mechanism for grouping related items into a functional unit for storing, displaying, and manipulating records.
- Items: These are interface objects that present data values to the user or enable the user to interact with the form, depending upon the item type. There are several different types of items. Items are logically grouped into *blocks* and visibly arranged on *canvases.*
- Canvases: A canvas is a “surface” where visual objects, such as graphics and items, are arranged. A form module can have several canvases (like the pages of a paper form). A canvas can display items from one or more blocks. To see a canvas and its items, you must display the canvas in a window. By default, all canvases in a form appear in the same window (which could mean you see only one canvas at a time), but you can assign separate windows for each canvas so that several canvases can be viewed at once.

Note: Items in one block do not need to be physically grouped. They can span many canvases (and windows).

Instructor Note

A canvas is like a picture portrait, and a window is like a picture frame. Just as you need a picture frame to display a picture portrait, you need a window to display a canvas and its contents.



Navigation in a Block

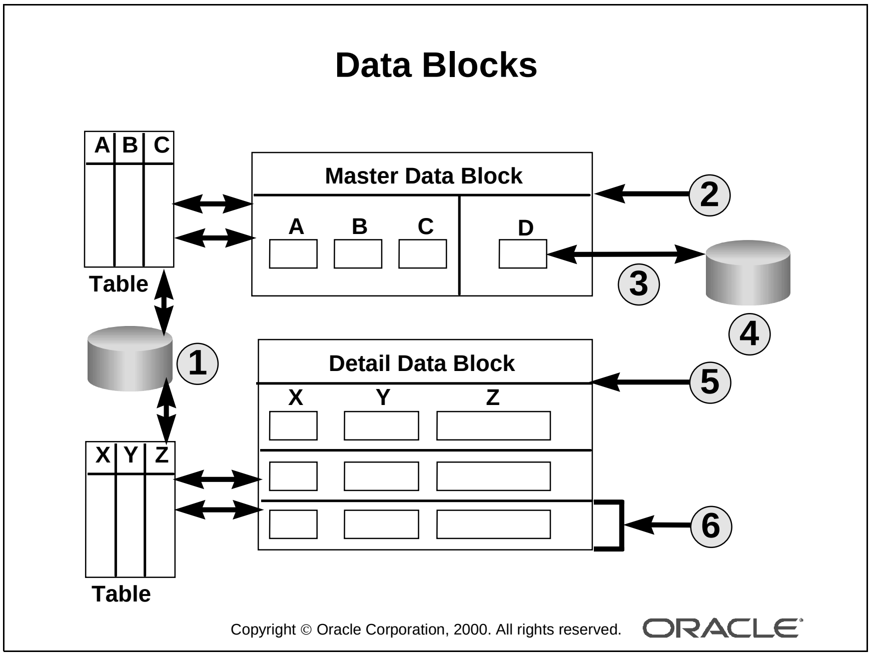
Copyright © Oracle Corporation, 2000. All rights reserved.

Navigation in a Form Module

When you run a form, you principally navigate around its contents by way of items and blocks, *not* by canvases. Each item has a sequenced position within its block, and each block has a sequenced position in the form.

When a user requests to move to the next item in a block, focus will be set on the next item in sequence, wherever that may be. If the next item is on a different canvas, Oracle Forms Developer displays that canvas automatically. Similarly, users can request to move to the next block (or previous block). If the first item in this block resides on another canvas, then that canvas is displayed automatically.

Of course, if you can already *see* the item that you want to move to, then you may click on it directly with the mouse. You can also program mechanisms into the application to enable navigation in other ways.



Data Blocks

Copyright © Oracle Corporation, 2000. All rights reserved.

1	Base table source
2	Single-record data block
3	Trigger access
4	Nonbase table source
5	Multirecord data block
6	Record

Types of Blocks

In Form Builder there are two main types of blocks: data blocks and control blocks.

Data Blocks When you build database applications with Form Builder, many of the blocks will be *data* blocks. A data block is associated with a specific database table (or view), a stored procedure, a FROM clause query, or transactional triggers.

If it is based on a table (or view), the data block can be based on only *one* base table, even though the data block can be programmed to access data from more than one table and data sources. By default, the association between a data block and the database enables the user to automatically access and manipulate data in the database. However, to access data from other tables (nonbase tables), you need to write triggers.

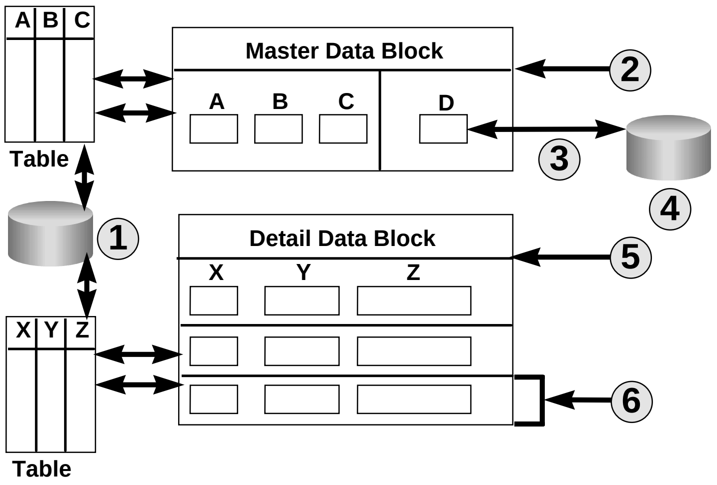
For a base table, Form Builder automatically performs the following actions:

- Creates items in the data block to correspond to columns in the table (These items are data items or base table items.)
- Produces code in the form to employ the rules of the table's constraints
- Generates SQL at run time (implicit SQL) to insert, update, delete, and query rows in the base table, based upon the user's actions

At run time, you can use standard function keys, buttons, menu options, or standard toolbar options to initiate query, insert, update, or delete operations on base tables, and the subsequent commit of the transaction.

Control Blocks A control block is not associated with a database, and its items do not relate to any columns within any database table. Its items are called control items. For example, you can create many buttons in your module to initiate certain actions and to logically group these buttons in a control block.

Data Blocks



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

1	Base table source
2	Single-record data block
3	Trigger access
4	Nonbase table source
5	Multirecord data block
6	Record

Master Versus Detail Blocks

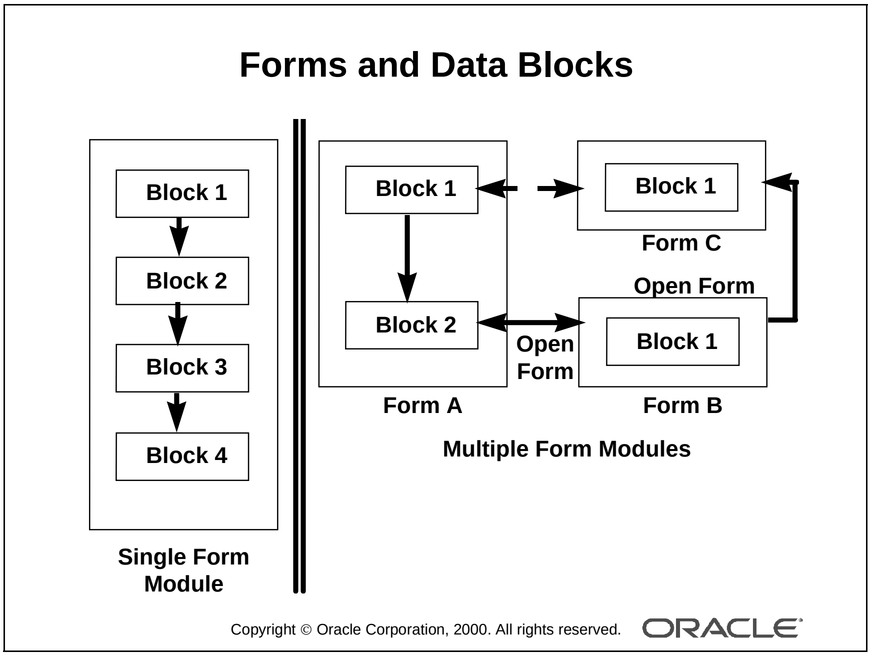
To support the relationship between data blocks and their underlying base tables, you can define one data block as the detail (child) of a master (parent) data block. This links primary key and foreign key values across data blocks, and synchronizes the data that these data blocks display.

Form Builder automatically generates the objects and code needed to support master-detail relationships. As the designer, you need only request it.

Note: If your application requires it, you can also create independent data blocks in which there is no relationship between the two data blocks.

Single-Record Versus Multirecord Blocks

You can design a data block to show one record at a time (single-record block) or several records at once (multirecord block). Usually, you create a single-record data block to show master block data and a multirecord data block to show detail block data. In either case, records in a data block that are currently not visible on the screen are stored in a block buffer.



Forms and Data Blocks

Block 1
Block 2
Block 3
Block 4

Single Form Module

Block 1
Block 2
Form A

Open Form

Block 1
Form C

Open Form

Block 1
Form B

Multiple Form Modules

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

This slide illustrates multiple data blocks in a single form compared to the multiple form application.

Many Blocks or Many Forms?

Typically, a Form Builder application consists of more than one data block. With more than one data block, you can do the following:

- Separate the navigation cycle of one group of items from another
- Map each data block to a different database table (You can have one base table per data block.)
- Produce a master-detail form, with a master data block and corresponding detail data blocks that are related to the master

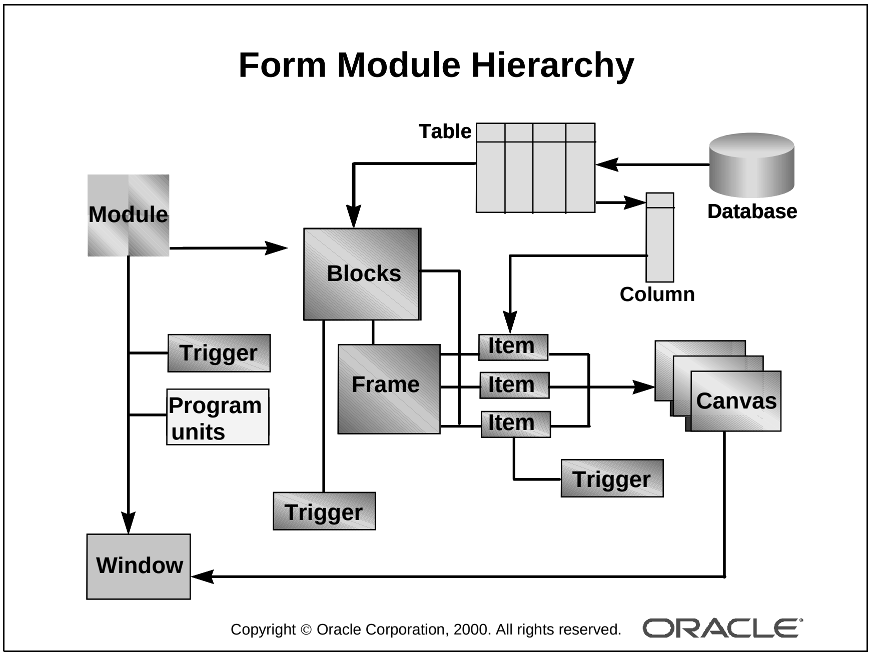
You can create a large form module with many data blocks. Alternatively, you can create several smaller form modules with fewer data blocks in each.

Generally, a modular application with several smaller form modules has the following characteristics:

- Modules are loaded only when their components are required, thus conserving memory.
- Maintenance can occur on one module without regenerating or loading the others.
- Forms can call upon one another, as required.
- Parallel development can be carried out by different team members on different components.

Here are some points to consider when grouping data blocks in the application:

Data Blocks in the Same Form Module	**Data Blocks in Different Form Modules**
The data blocks can be directly linked in master-detail relationships.	The data blocks cannot be linked by the standard interblock relations.
Navigation between data blocks is handled by default functionality.	Navigation between data blocks of different forms is programmed by the designer (although mouse navigation to visible items can be automatic).



Form Module Hierarchy

Copyright © Oracle Corporation, 2000. All rights reserved.

Technical Note

A form module is made up of one or more blocks. A data block is based on a database object, such as a table or a view. A data block can contain both data items and control items. A frame can be created to arrange data block items. Each item in a block must appear on a canvas, and each canvas must appear in a window. A form module can have one or more canvases and windows.

Using triggers, you can add functionality to your form. Triggers can be written at different levels in a form module. User-named program units enable you to write additional PL/SQL code through procedures, functions, and packages.

The Object Hierarchy

You can create many types of objects in a form module. They are discussed in more detail in later lessons.

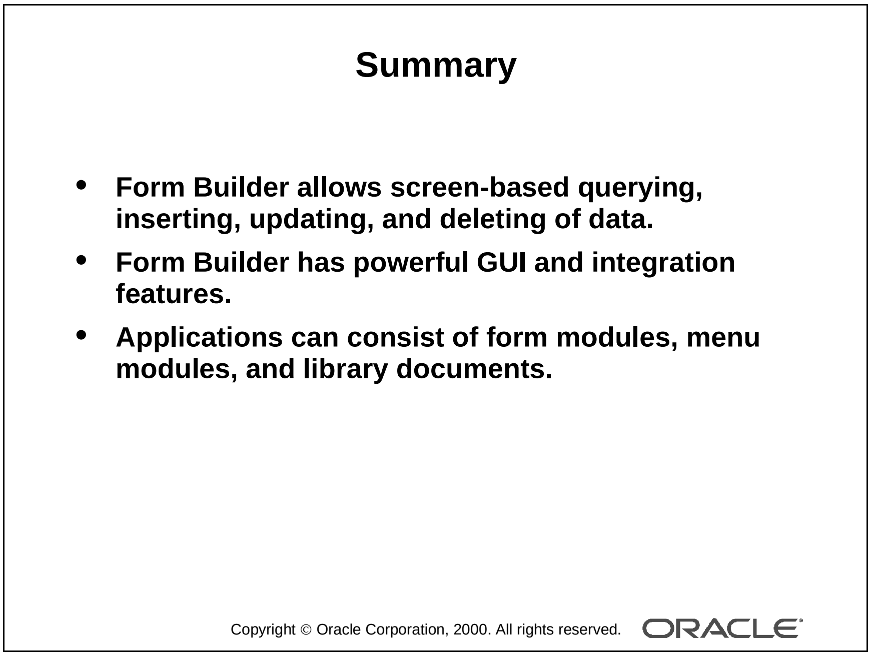
In the following table, note that some objects are associated, even though one might not be “owned” by the other.

Object	**Description**
Data block	Logical section of a form; owned by the form module
Item	Member of a data block (Items are functionally grouped into records.)
Canvas	The surface where visual objects are arranged; owned by the form module A canvas can contain text and graphics—static information that the user cannot interact with.
Window	Produced to contain the views of canvases; owned by the form module
Frame	A graphic object that appears on a canvas (A frame is used to arrange the items within a data block and is owned by the canvas it appears on.)
User-named program unit	Named procedure, function, or package; owned by the form module
Trigger	PL/SQL block executed on an event (Triggers can be owned by the form module, a data block, or an item, depending upon their scope.)
Other objects	Include alerts, parameters, and record groups (These are mainly owned by the form module itself.)

The structure of a menu module is discussed in the course *Oracle 9iDS Forms: Build Internet Applications II*.

Instructor Note

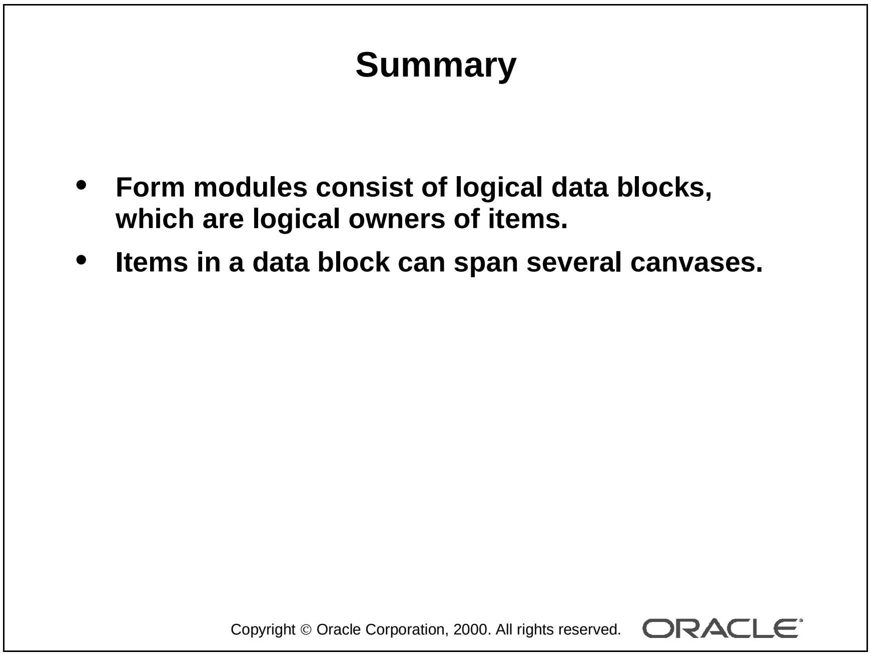
Point out the above objects to the students, using the Object Navigator.



Summary

- **Form Builder allows screen-based querying, inserting, updating, and deleting of data.**
- **Form Builder has powerful GUI and integration features.**
- **Applications can consist of form modules, menu modules, and library documents.**

Copyright © Oracle Corporation, 2000. All rights reserved.



Summary

- **Form modules consist of logical data blocks, which are logical owners of items.**
- **Items in a data block can span several canvases.**

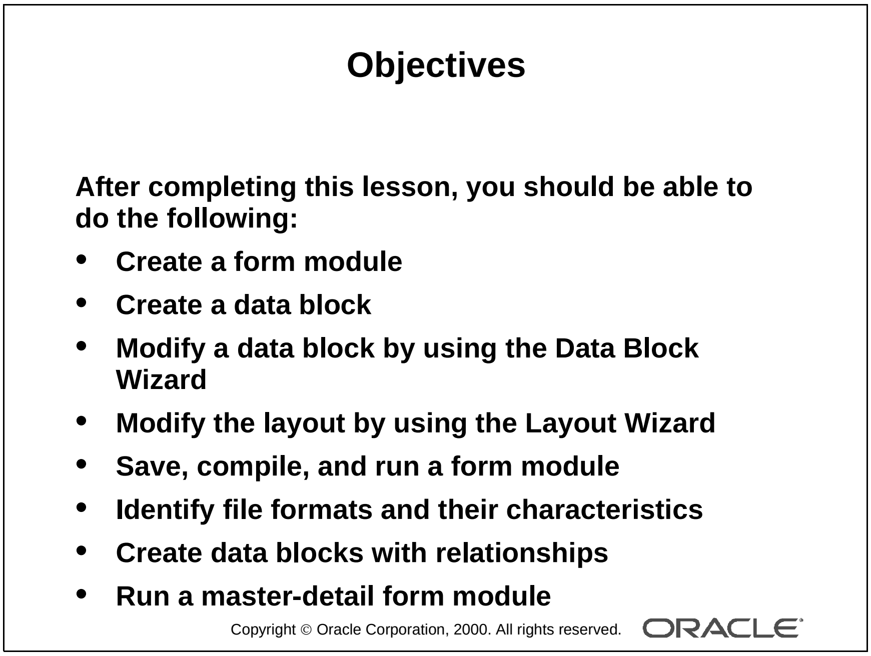
Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

- With Form Builder, an Oracle Forms Developer component, you can develop form-based applications for presenting and manipulating data in a variety of ways. Form Builder enables screen-based queries, inserts, updates, and deletes of data.
- Form Builder provides powerful GUI and integration features.
- Applications consist of form modules, menu modules, and library documents. A form presents the objects and data with which users can see and interact. A menu module can consist of a hierarchy of menus, each with selectable items. A library document is a collection of PL/SQL program units. The use of the Object Library can ensure consistency and help prevent developers from re-creating objects that have already been developed.
- Form modules consist of logical data blocks. A data block is the logical owner of items.
- Items in one data block do not need to be physically grouped. Items in one data block can span several canvases.

4

Creating a Basic Form Module



Objectives

After completing this lesson, you should be able to do the following:

- **Create a form module**
- **Create a data block**
- **Modify a data block by using the Data Block Wizard**
- **Modify the layout by using the Layout Wizard**
- **Save, compile, and run a form module**
- **Identify file formats and their characteristics**
- **Create data blocks with relationships**
- **Run a master-detail form module**

Copyright © Oracle Corporation, 2000. All rights reserved.

Schedule

Topic	**Timing**
Lecture	40 minutes
Practice Session 1	30 minutes
Lecture	40 minutes
Practice Session 2	30 minutes
Total	140 minutes

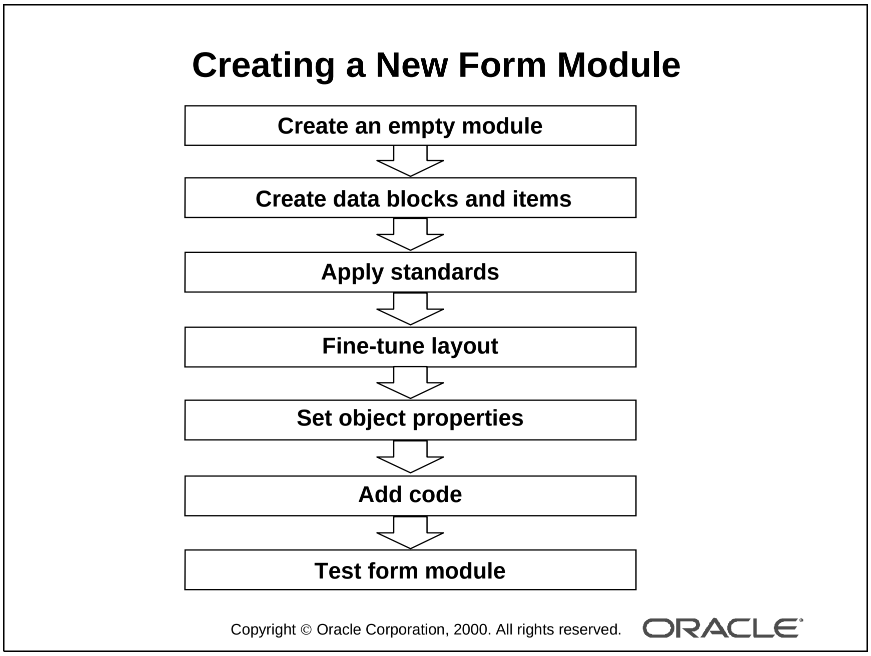
Instructor Note

This lesson is divided into two sections. There is a practice midway through the lesson, and another practice at the end of the lesson.

Introduction

Overview

Oracle Forms Developer applications usually consist of a number of form modules. Each form module consists of data blocks that are built using table specifications from the database. This lesson shows you how to create a basic form module and its data blocks.



Creating a New Form Module

Create an empty module

Create data blocks and items

Apply standards

Fine-tune layout

Set object properties

Add code

Test form module

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a New Form Module

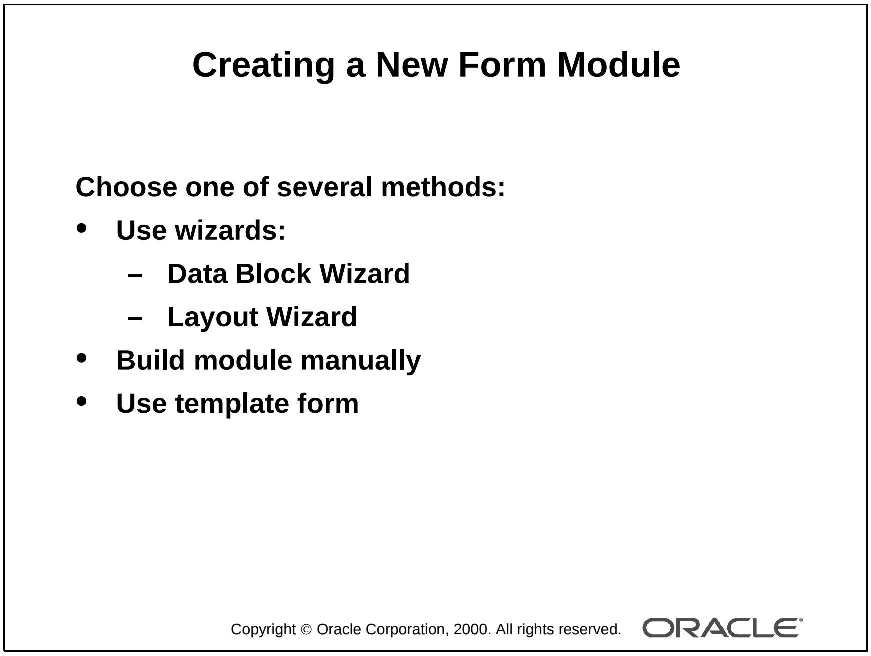
This lesson covers the basic process for creating a new form module and data blocks within it.

How to Create a New Form Module

1. Create an empty form module.
2. Create data blocks and items.
3. Apply user interface standards to objects.
4. Fine-tune the layout.
5. Set object properties.
6. Add code.
7. Test the form module.

Creating a New Form Module

To Do This	Use This Form Builder Tool
Create an empty module	Object Navigator
Create data blocks and items	Data Block Wizard
Apply standards	Object Library
Fine-tune the layout	Layout Wizard or Layout Editor
Set object properties	Property Palette
Add code	PL/SQL Editor
Test the form module	Forms Runtime



Creating a New Form Module

Choose one of several methods:

- **Use wizards:**
  - **Data Block Wizard**
  - **Layout Wizard**
- **Build module manually**
- **Use template form**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Create a new, empty form module and change its name.

Methods for Creating a New Form Module

You can create a new form module in many different ways.

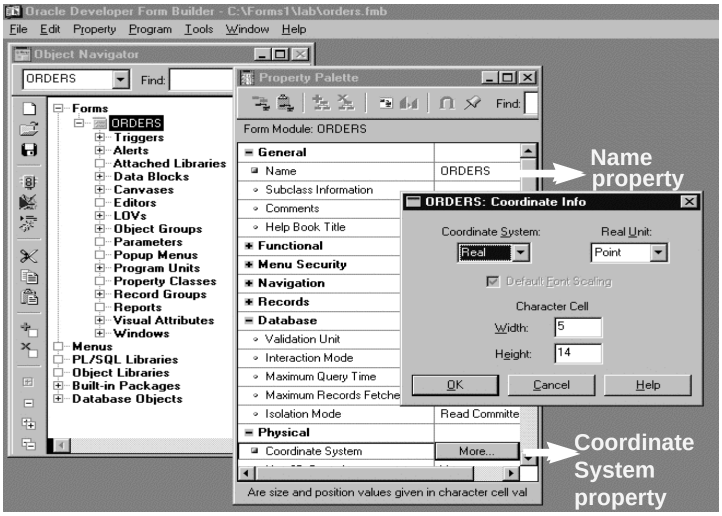
- Invoke the Form Builder component. This takes you to the Form Builder Welcome page. Now do one of the following:
  - Select the “Use the Data Block Wizard” option, then follow the required data block creation steps. Then follow the Layout Wizard steps.
  - Select the “Build a new form manually” option. This takes you into the Form Builder Object Navigator (automatically creating an empty form module).
  - Select the “Build a form based on a template” option and use a template form.
- If you are already in the Form Builder component, you can create a new form module by doing one of the following:
  - Double-click the Forms node in the Object Navigator (only when no other form modules are available).
  - Select File—>New—>Form.
  - Select the Object Navigator node for Forms, and then click the Create icon.

Changing the Form Module Name

- When you first build a form module, Form Builder assigns the name MODULE*XX* to the new form module, where *XX* is the next number available for module names. This name is displayed in the Object Navigator and in the Property Palette. You should change the default name to a meaningful name in either of the following places:
- In the Object Navigator:
  - Double-click the form module name.
  - Change the default name as desired and press [Enter].
- In the Property Palette (shown on the next page)

Note: Follow Oracle naming rules. Do not give two objects of the same type the same name. The name cannot include Oracle or Form Builder reserved words.

Form Module Properties



Form Module: ORDERS

General	
Name	ORDERS
Subclass Information	
Comments	
Help Book Title	
Functional	
Menu Security	
Navigation	
Records	
Database	
Validation Unit	
Interaction Mode	
Maximum Query Time	
Maximum Records Fetche	
Isolation Mode	Read Committe
Physical	
Coordinate System	More...

Are size and position values given in character cell val

ORDERS: Coordinate Info

Coordinate System: Real

Real Unit: Point

☑ Default Font Scaling

Character Cell

Width: 5

Height: 14

Name property

Coordinate System property

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Open the Property Palette of the new form module. Concentrate only on the Name and Coordinate System properties. Point out that other properties will be discussed in later lessons. Open the Coordinate Info window and explain the implications of choosing the units. (72 points = 1 inch). Points allow for flexibility when designing cross platform applications.

Setting Form Module Properties

Each form module consists of several objects. Objects within a form, and the form module itself, have properties that define their behavior. You can see the properties of an object and their values in its Property Palette.

To open the Property Palette of an object, do one of the following:

- Double-click the object’s icon in the Object Navigator.
- Select the object in the Object Navigator and select Tools—>Property Palette.

Define the properties of the form module when you first create it. The properties affect the general behavior of the form and the objects within it. Properties for a form module include the following:

Property	**Description**
Name	Specifies the internal name of the form module, as it appears in the Object Navigator
Coordinate System	Defines the units used to measure objects in the form and their positions (see next section)

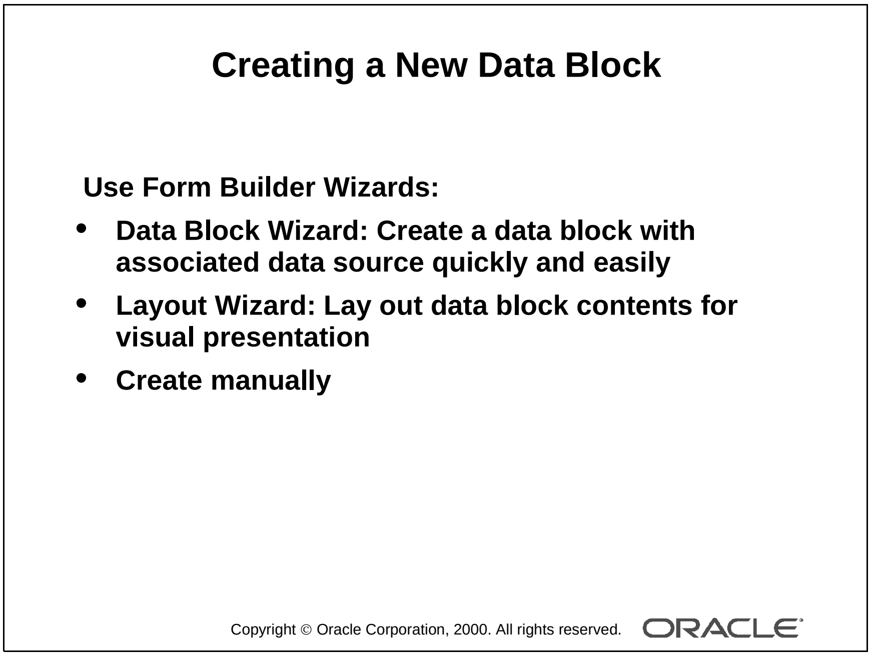
Choosing a Unit for the Coordinate System

When you click More in the Property Palette window with the Coordinate System property selected, the Coordinate Info window opens.

The Coordinate System unit for a form can be one of the following:

- Real
  - Unit can be pixel, centimeter, inch, point, or decipoint.
  - Real units are suitable for GUI applications and enable flexibility and fine alignment when adjusting object positions and sizes.
- Character
  - Units are character cells (default size taken from the default font settings).
  - Character units are suitable where the run-time environment includes character mode devices.

The default unit is point (Real). This means that object positions and sizes within the form are measured by this unit. Points provide fine alignment and consistency across different platforms and video devices.

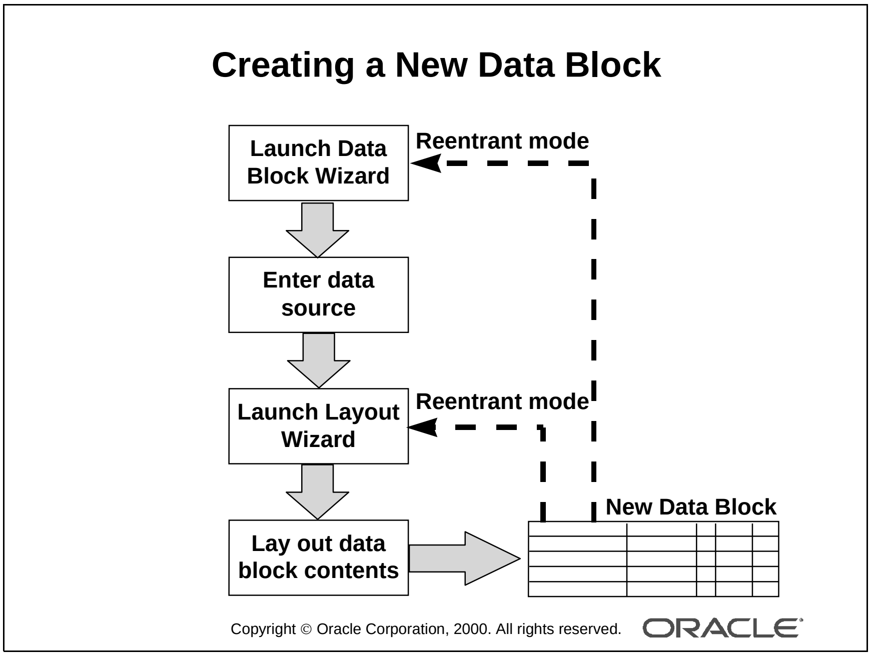


Creating a New Data Block

Use Form Builder Wizards:

- **Data Block Wizard: Create a data block with associated data source quickly and easily**
- **Layout Wizard: Lay out data block contents for visual presentation**
- **Create manually**

Copyright © Oracle Corporation, 2000. All rights reserved.



Creating a New Data Block

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a New Data Block

A form module consists of one or more data blocks and control blocks. Now that you know how to create a new form module, you need to create new data blocks within it.

Block creation involves creating the data block and then laying out its contents for visual presentation. You can create a data block manually or by using the Form Builder wizards. In this lesson you learn how to create a new data block based on a database table, using the Data Block Wizard and the Layout Wizard.

Note: Recall that a data block can be based on a table or view, a stored procedure, a FROM clause query, or a transactional trigger. In this course, you use database tables as the source; other sources for creating data blocks are covered in the course *Oracle 9iDS Forms: Build Internet Applications II*.

Data Block Wizard

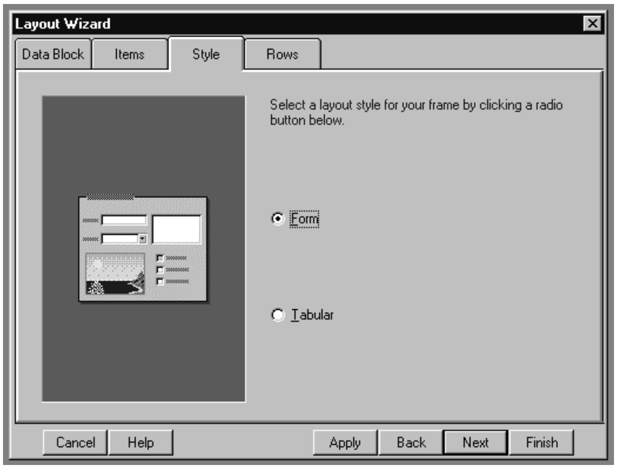
The Data Block Wizard enables you to create (or modify) data blocks quickly and easily for use in your application. The wizard can automatically generate code to enforce integrity constraints in the database.

Layout Wizard

Although the Data Block Wizard allows you to create a new data block easily with its associated data sources, it does not deal with the visual presentation of objects included in the data block. Once you create the data block, you need to lay out its contents for user interaction. To accomplish this task quickly and easily, use the Layout Wizard.

Note: The wizards are not the only way to perform a task such as building a data block, but they are usually the simplest. You can build a block manually instead of using the wizards.

Navigating the Wizards



Select a layout style for your frame by clicking a radio button below.

● Form

○ Tabular

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

If you click the Next or Back buttons before entering all necessary information for a particular wizard page, the wizard prevents you from navigating to another page. Similarly, if you have not entered all necessary information into the wizard when you click Apply or Finish, the wizard automatically takes you to the page where you can finish entering the required information.

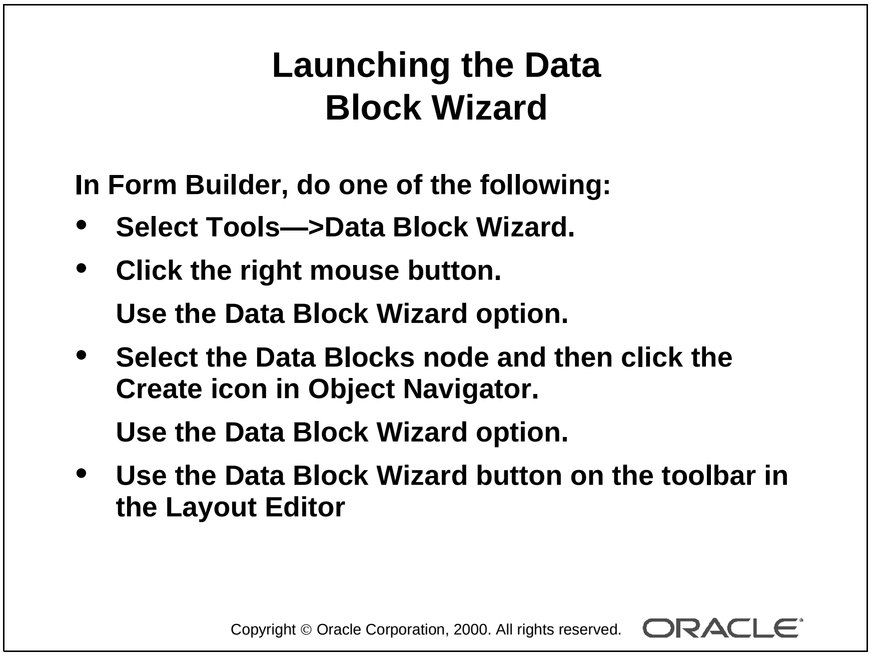
The reason for showing a wizard page in reentrant mode is to illustrate all the buttons. The [Apply] button is only available in reentrant mode.

Navigating the Wizards

The Data Block Wizard and the Layout Wizard provide several buttons to help you navigate:

Button	**Description**
Cancel	Cancels any changes and exits the wizard
Help	Displays online help text for the current wizard page
Back	Navigates to the previous page in the wizard
Next	Navigates to the next page in the wizard
Apply	Applies your changes without exiting the wizard (available only upon reentering the wizard)
Finish	Saves any changes and exits the wizard

Note: Apply is available only when you reenter the Layout Wizard to modify an existing layout.



Launching the Data Block Wizard

In Form Builder, do one of the following:

- **Select Tools—>Data Block Wizard.**
- **Click the right mouse button.**

  Use the Data Block Wizard option.
- **Select the Data Blocks node and then click the Create icon in Object Navigator.**

  Use the Data Block Wizard option.
- **Use the Data Block Wizard button on the toolbar in the Layout Editor**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

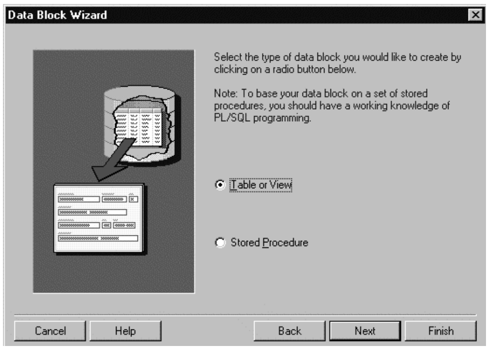
Demonstration Create a new data block by using the Form Builder wizards. Base the data block on the S_ORD table.

Launching the Data Block Wizard

Launch the Data Block Wizard by doing one of the following:

- In Form Builder, do one of the following:
  - Select Tools—>Data Block Wizard from the Form Builder default menu system.
  - Click the right mouse button and select the Data Block Wizard option.
  - In the Object Navigator, select the Data Blocks node, then click the Create icon. In the New Data Block dialog box, select the Use the Data Block Wizard option.
  - In the Layout Editor, click the Data Block Wizard button on the toolbar.
- If you are not already in Form Builder, launch Form Builder and select the Use the Data Block Wizard option in the Form Builder Welcome page.

Data Block Wizard: Type Page



Select the type of data block you would like to create by clicking on a radio button below.

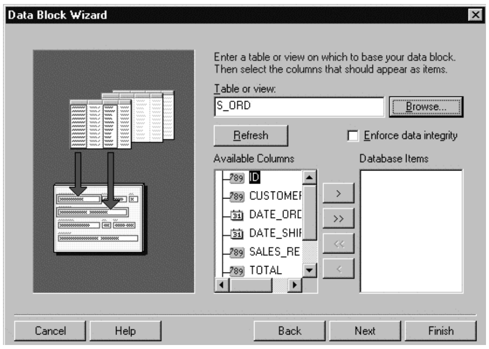
Note: To base your data block on a set of stored procedures, you should have a working knowledge of PL/SQL programming.

● Table or View

○ Stored Procedure

Copyright © Oracle Corporation, 2000. All rights reserved.

Data Block Wizard: Table Page



Enter a table or view on which to base your data block. Then select the columns that should appear as items.

Table or view: S_ORD

□ Enforce data integrity

Available Columns

- ID
- CUSTOMER
- DATE_ORD
- DATE_SHIP
- SALES_RE
- TOTAL

Database Items

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

You can also select Oracle8 objects as object data sources in the Layout Wizard. Oracle8 objects as data sources are covered in the *Build Internet Applications II* course. Appendix E of this course describes the basics of Oracle8 objects.

Creating a New Data Block with the Data Block Wizard

Use the Data Block Wizard to create a new data block with its associated data sources. The Data Block Wizard consists of several pages. To create a new data block, you must interact with each page.

Type Page Choose between one of two data source types:

- Table or View
- Stored Procedure

Select the Table or View (default) option.

Table Page

1. Enter the table or view name for the data source name, or click Browse and select a name from a dialog box.
2. Click Refresh to display a list of columns in the selected table or view.

   If you are not connected to the database, the Connect box is displayed.
3. Select the columns you want to include in the data block. (Use [Control]-click to select more than one column.)
4. Click >> or << to include or exclude all columns, or click > or < to include or exclude selected columns only. You can also drag and drop selected columns from one list to another.
5. Select the Enforce Data Integrity check box if you want the wizard to enforce the database integrity constraints.

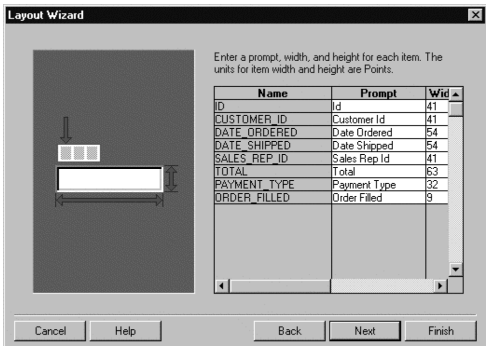
Note: If there is at least one other existing block in the current module, you next get the Master-Detail page, where you can associate the new data block with other master data blocks. This page is discussed later in the lesson.

Finish Page Select the “Create the data block, then call the Layout Wizard” option. Select Finish to create the new data block and immediately invoke the Layout Wizard.

Note: You have the option of exiting the Data Block Wizard at this stage, without immediately invoking the Layout Wizard. If you do so, you can either lay out the data block manually or invoke the Layout Wizard at a later time to lay out the items of a data block.

To invoke the Layout Wizard at a later time, select the data block in the Object Navigator, and choose Tools—>Layout Wizard.

Layout Wizard: Items Page

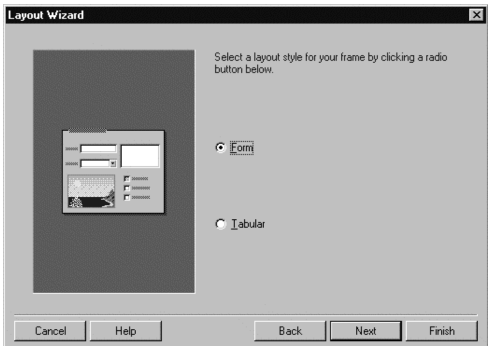


Enter a prompt, width, and height for each item. The units for item width and height are Points.

Name	Prompt	Wid
ID	Id	41
CUSTOMER_ID	Customer Id	41
DATE_ORDERED	Date Ordered	54
DATE_SHIPPED	Date Shipped	54
SALES_REP_ID	Sales Rep Id	41
TOTAL	Total	63
PAYMENT_TYPE	Payment Type	32
ORDER_FILLED	Order Filled	9

Copyright © Oracle Corporation, 2000. All rights reserved.

Layout Wizard: Style Page



Select a layout style for your frame by clicking a radio button below.

● Form

○ Tabular

Copyright © Oracle Corporation, 2000. All rights reserved.

Laying Out a New Data Block with the Layout Wizard

Use the Layout Wizard to lay out the data block items for visual presentation quickly and easily. The Layout Wizard consists of several pages. You must interact with each page.

Welcome Page Click Next to continue.

Canvas Page

1. Select New Canvas from the Canvas pop-up list to get a new canvas on which to display the data block items.
2. Select Content as the canvas type in the Type pop-up list.

Data Block Page

1. Select the items you want to display in the data block frame. (Use [Ctrl]-click to select more than one column.)
2. Click >> or << to include or exclude all items, or click > or < to include or exclude selected items only. You can also drag and drop selected items from one list to another.

   Note: To lay out the items in a particular sequence, drag and drop items into that sequence.
3. You can use the Item Type pop-up list to select a type for each item. The default type is Text for each item.

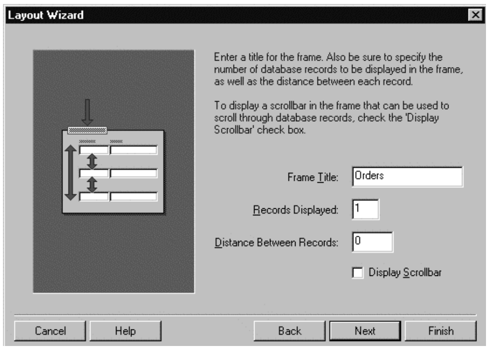
   Note: An item type can also be changed later to something else, such as pop-up list or radio group.

Items Page Specify the prompt text and the display width and height for each display item for the data block.

Style Page Select a layout style for your frame. Your options are:

- Form (usually used to create single-record data blocks)
- Tabular (usually used to create multirecord data blocks)

Layout Wizard: Rows Page



Layout Wizard

Enter a title for the frame. Also be sure to specify the number of database records to be displayed in the frame, as well as the distance between each record.

To display a scrollbar in the frame that can be used to scroll through database records, check the 'Display Scrollbar' check box.

Frame Title: Orders

Records Displayed: 1

Distance Between Records: 0

□ Display Scrollbar

Copyright © Oracle Corporation, 2000. All rights reserved.

Data Block Functionality

Once you create a data block with the wizards, Form Builder automatically creates:

- **A form module with database functionality including query, insert, update, delete**
- **A frame object**
- **Items in the data block**
- **A prompt for each item**

Copyright © Oracle Corporation, 2000. All rights reserved.

Laying Out a New Data Block with the Layout Wizard (continued)

Rows Page

1. Enter a title in the Frame Title field.
2. Enter the number of records you want to display at run time in the Records Displayed field.
3. Enter the physical distance (in the coordinate system unit of the form) between records.
4. You can select the Display Scrollbar check box to display a scroll bar next to the frame (common for multirecord data blocks).

Finish Page Select Finish to create a new frame and lay out the selected items for the new data block. The Layout Wizard steps are complete.

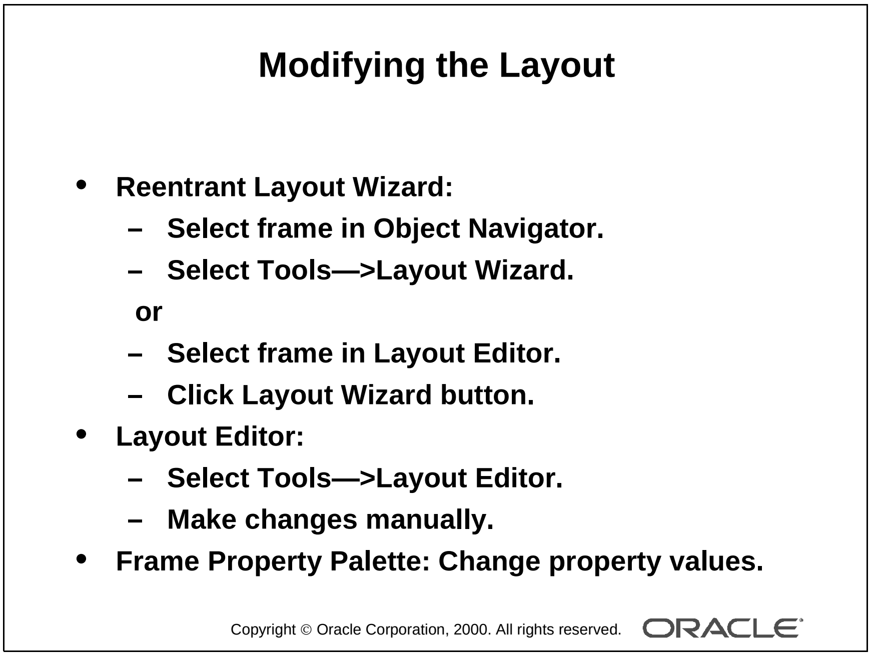
Note: Once you complete the Layout Wizard steps, you can view the layout in the Layout Editor. Here you can customize or modify the layout if necessary.

Data Block Functionality

Once you create a new data block by using the wizards, Form Builder *automatically* creates the following objects for you:

- A new form module with a default menu (Basic database functionality such as querying, inserting, updating, and deleting is automatically available on the items in the base table block when you run the new form.)

  The new data block is created with default property values. These values can be modified to change the behavior of the form.
- A frame object to arrange the items within the new data block
- An item for each database table column included in the data block (Each item is assigned default property values to match the underlying column specifications.)
- A prompt for each item in the data block (The default prompt is the name of the column.)



Modifying the Layout

- **Reentrant Layout Wizard:**
  - **Select frame in Object Navigator.**
  - **Select Tools—>Layout Wizard.**

  or

  - **Select frame in Layout Editor.**
  - **Click Layout Wizard button.**
- **Layout Editor:**
  - **Select Tools—>Layout Editor.**
  - **Make changes manually.**
- **Frame Property Palette: Change property values.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Invoke the Layout Wizard in reentrant mode to show how to modify the layout of the new data block created in the previous demonstration. Make sure you select the correct frame before invoking the Layout Wizard.

Modifying the Layout

Once you create a data block, you may want to customize or modify its layout. You can do this by doing one of the following:

- Reenter the Layout Wizard (see the next section), and use it to make the changes.
- Select Tools—>Layout Editor to invoke the Layout Editor and make changes manually in the editor.
- Change the property values of the frame in its Property Palette.

Invoking the Layout Wizard in Reentrant Mode

A very powerful feature of the Layout Wizard is its ability to operate in *reentrant* mode. Use the reentrant mode to modify the layout of items in an existing frame, even if the frame was not originally created with the Layout Wizard.

Invoke the Layout Wizard in reentrant mode from the Object Navigator or the Layout Editor.

- From the Object Navigator:
    - Select the appropriate frame (under the Canvases node).
    - Select Tools—>Layout Wizard.

  or

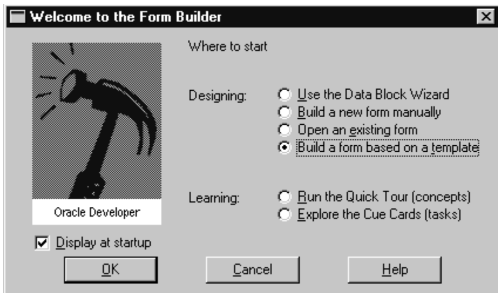
  Click the right mouse button and select the Layout Wizard option.
- In the Layout Editor:
    - Select the appropriate frame.
    - Click Layout Wizard.

  Note: Before you reenter the Layout Wizard, it is important to select the correct frame in the Object Navigator or the Layout Editor. If you overlook this when you reenter the Layout Wizard, you may create an additional frame instead of modifying the current frame.

Either method takes you to the Data Block page in the Layout Wizard. Use the Next and Back buttons as before, or go directly to a certain page by clicking the particular page tab.

Note: If necessary, you can also invoke the Data Block Wizard in reentrant mode to modify an existing data block. To do so, select the existing data block in the Object Navigator, and choose Tools—>Data Block Wizard.

Template Form



Welcome to the Form Builder

Where to start

Designing:
○ Use the Data Block Wizard
○ Build a new form manually
○ Open an existing form
● Build a form based on a template

Learning:
○ Run the Quick Tour (concepts)
○ Explore the Cue Cards (tasks)

☑ Display at startup

Copyright © Oracle Corporation, 2000. All rights reserved.

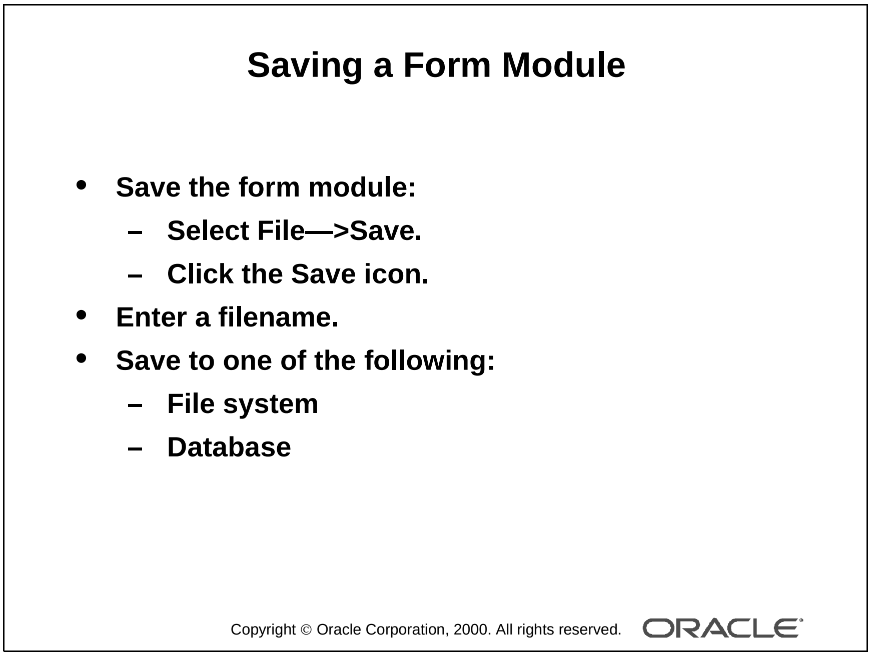
Template Forms

You can create a new form based on standard template forms, so that you can provide other team members with a default starting point. Templates typically include generic objects, such as graphics, toolbars, and program units. You can define standard window layouts, standard toolbars, and other common objects that you want to include in new forms.

Creating a Form Based on a Template

To create a form based on a template, follow these steps:

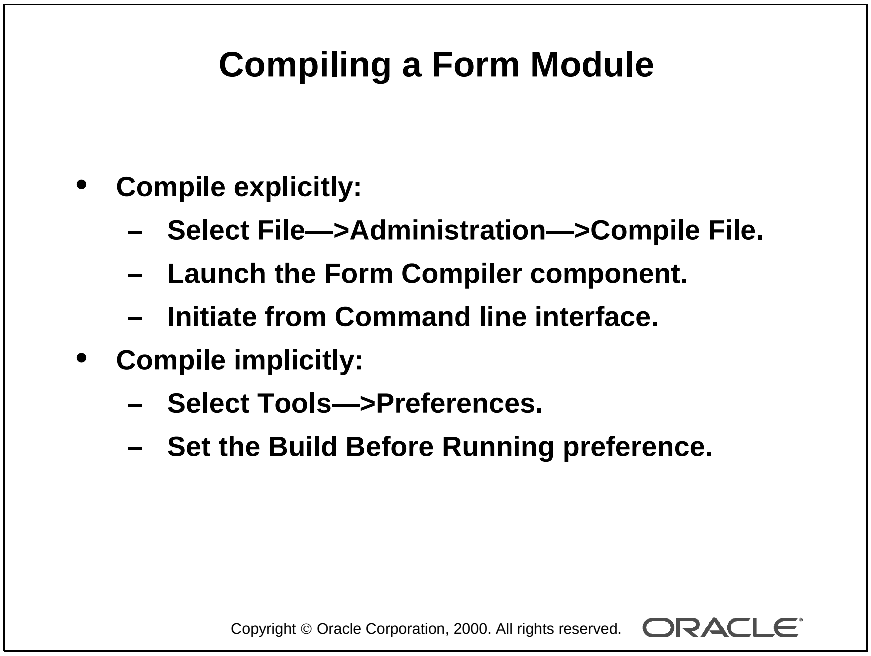
1. Start Form Builder.
2. In the Welcome to the Form Builder dialog box, select the “Build a form based on a template” option, and then click OK.



Saving a Form Module

- **Save the form module:**
  - **Select File—>Save.**
  - **Click the Save icon.**
- **Enter a filename.**
- **Save to one of the following:**
  - **File system**
  - **Database**

Copyright © Oracle Corporation, 2000. All rights reserved.



Compiling a Form Module

- **Compile explicitly:**
  - **Select File—>Administration—>Compile File.**
  - **Launch the Form Compiler component.**
  - **Initiate from Command line interface.**
- **Compile implicitly:**
  - **Select Tools—>Preferences.**
  - **Set the Build Before Running preference.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The file system icons for FMB and FMX files have changed in Release 6*i*. Note that the icon for FMX files has a red dot in the lower right corner.

Saving, Compiling, and Running a Form Module

Saving a Form Module

You can save the form module definition by doing one of the following:

- Select File—>Save.
- Select the Save icon.

Both of these options display the File dialog box for the initial save. In the dialog box, do the following:

1. Enter a filename.
2. Click OK.

Note: You can choose to save a form module to either the database or the file system by setting your preference after selecting Tools—>Preferences. If you save to the file system, an `.fmb` file is produced. This *saved* definition of a form in the file system or database is not executable.

Note: When you work with more than one module at a time, Form Builder keeps track of the changes you make to each module separately. When you issue a Save command, only the current module is saved.

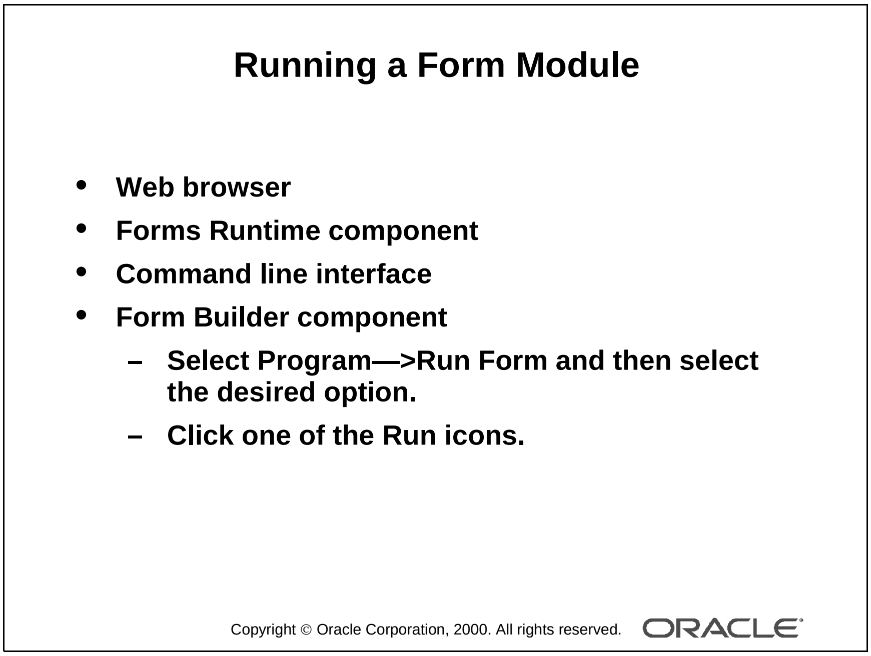
Compiling a Form Module

Before you can run a form, you must compile an executable (`.fmx)` file from the design (`.fmb`) file you created in Form Builder. Compiling a form (or menu) module creates the needed executable file.

- You can compile a form *explicitly* by doing one of the following:
  - Selecting File—>Administration—>Compile File after opening the form module
  - Launching the Form Compiler component
  - Initiating the process from the command line interface
- You can compile a form *implicitly* or automatically by checking the Build Before Running preference.

  Form Builder implicitly compiles the form module when you attempt to run it. Set this preference by selecting Tools—>Preferences.

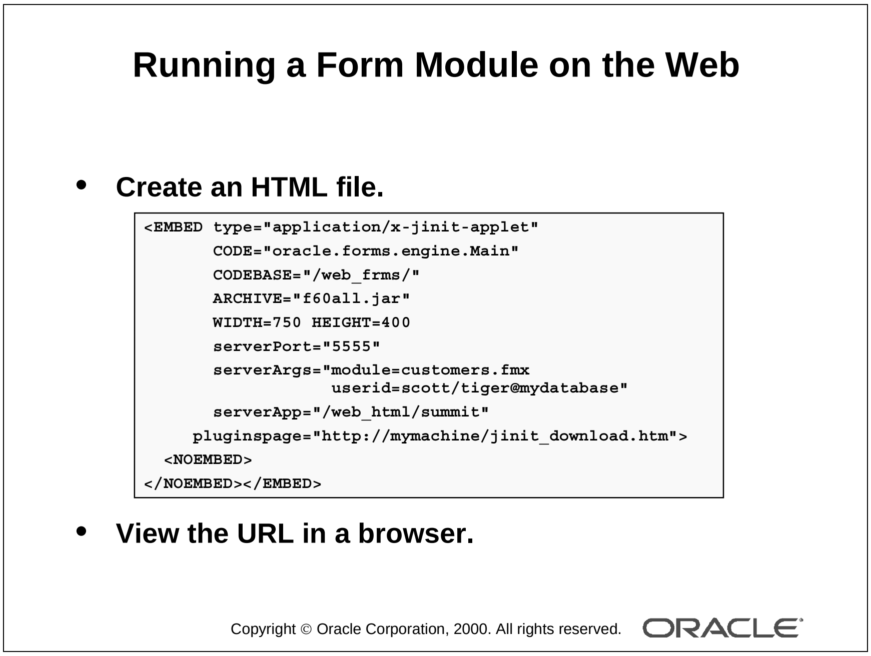
Note: Compiling and saving are two independent tasks. Performing one does not automatically accomplish the other. Both tasks must occur separately.



Running a Form Module

- Web browser
- Forms Runtime component
- Command line interface
- Form Builder component
  - Select Program—>Run Form and then select the desired option.
  - Click one of the Run icons.

Copyright © Oracle Corporation, 2000. All rights reserved.



Running a Form Module on the Web

- Create an HTML file.

```
<EMBED type="application/x-jinit-applet"
       CODE="oracle.forms.engine.Main"
       CODEBASE="/web_frms/"
       ARCHIVE="f60all.jar"
       WIDTH=750 HEIGHT=400
       serverPort="5555"
       serverArgs="module=customers.fmx
                   userid=scott/tiger@mydatabase"
       serverApp="/web_html/summit"
     pluginspage="http://mymachine/jinit_download.htm">
  <NOEMBED>
</NOEMBED></EMBED>
```

- View the URL in a browser.

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The Run Form Web button requires that Oracle Forms Services be installed on the client machine. That is not the recommended setup for this class.

Running a Form Module

You can run a form module from one of the following:

- Web browser
- Forms Runtime component
- Command line interface
- Form Builder component
  - Select Program—>Run Form and then one of the three Run Form options:

Run Form Option	Description
Run Form Client/Server	Use this option to run your form by using the client-server architecture.
Run Form Web	Use this option to run your form in the Web Previewer. Your form will look and behave as though being deployed from a Web server. The Web Previewer can be used to test your Web forms without having to actually deploy the forms on a server.
Run Form Debug	Use this option to invoke the debug mode for the Forms Runtime session.

  or

  - Click one of the three Run icons.

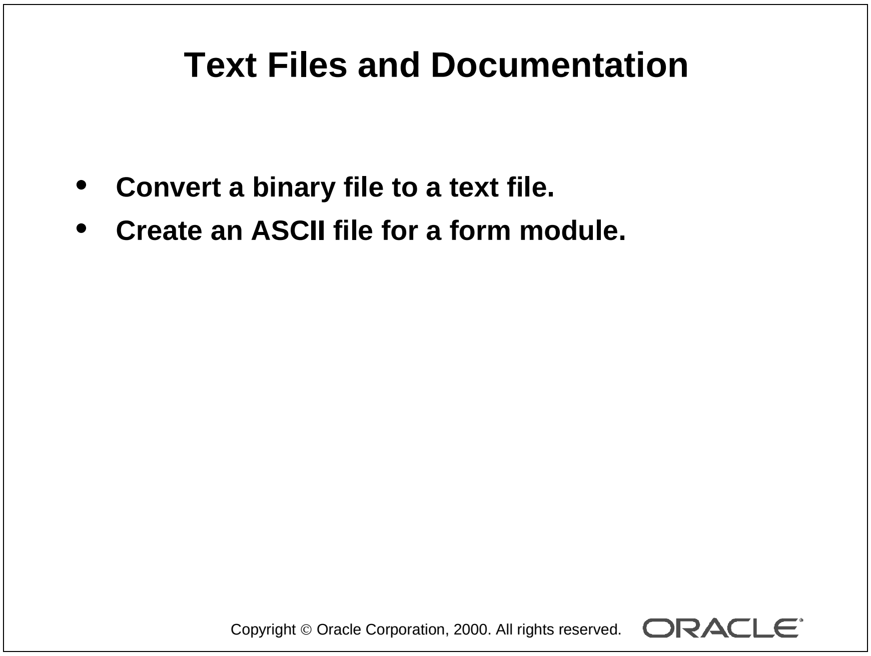
Note: If more than one form module is open in Form Builder, make the form module you want to run the current one by selecting any of its objects in the Object Navigator.

Running a Form Module on the Web

Once a standard Web server and Oracle Forms Services have been installed and configured, Forms applications can be run from a browser by using an HTML file. The HTML file specifies the Forms Server to connect to, the module name, the username, and other application-specific parameters. A sample HTML file is provided with Oracle Forms Services to use as a starting point for your own application.

To run the application, simply open your browser and specify the appropriate URL.

Note: Configuring a Web browser and Forms Services and creating an HTML file are covered in detail in the course *Oracle 9iAS Forms Services: Deploy Internet Applications.*



Text Files and Documentation

- Convert a binary file to a text file.
- Create an ASCII file for a form module.

Copyright © Oracle Corporation, 2000. All rights reserved.

Producing Text Files and Documentation

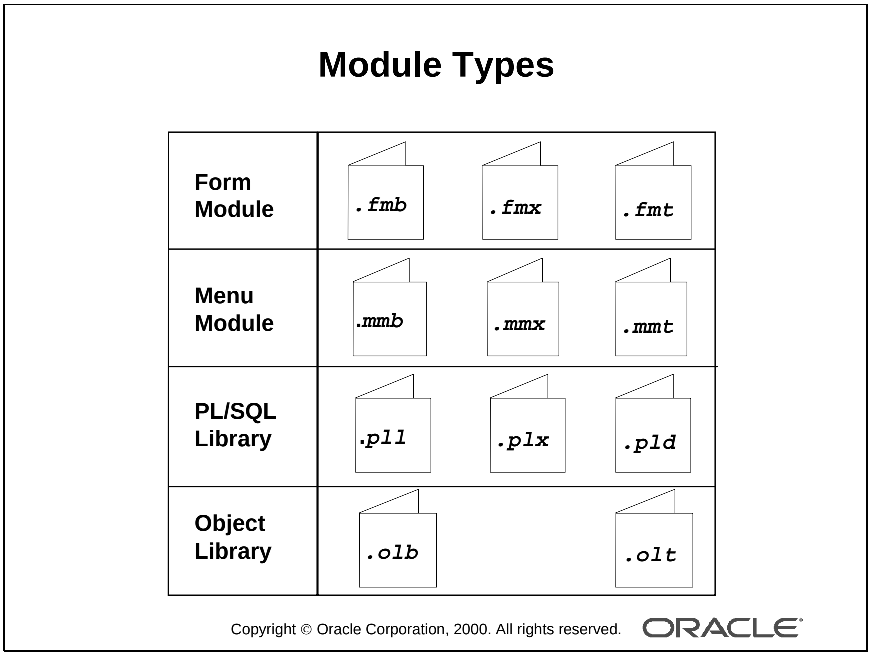
The files normally produced by saving and generating modules are in binary format. You can convert a binary file to text by doing the following:

1. Select File—>Administration—>Convert.

   This opens the Convert dialog box.
2. Select the type of module (Form, Menu, PL/SQL Libraries, Object Libraries), the file to convert, and the direction (Binary-to-Text).
3. Select Convert. This produces a text file for the module.

You can produce documentation for your module by doing the following:

1. Select the module to be documented in the Object Navigator.
2. Select File—>Administration—>Object List Report from the menu system. This produces an ASCII file with the name `<module>.txt`.



Module Types

Form Module	.fmb	.fmx	.fmt
Menu Module	.mmb	.mmx	.mmt
PL/SQL Library	.pll	.plx	.pld
Object Library	.olb		.olt

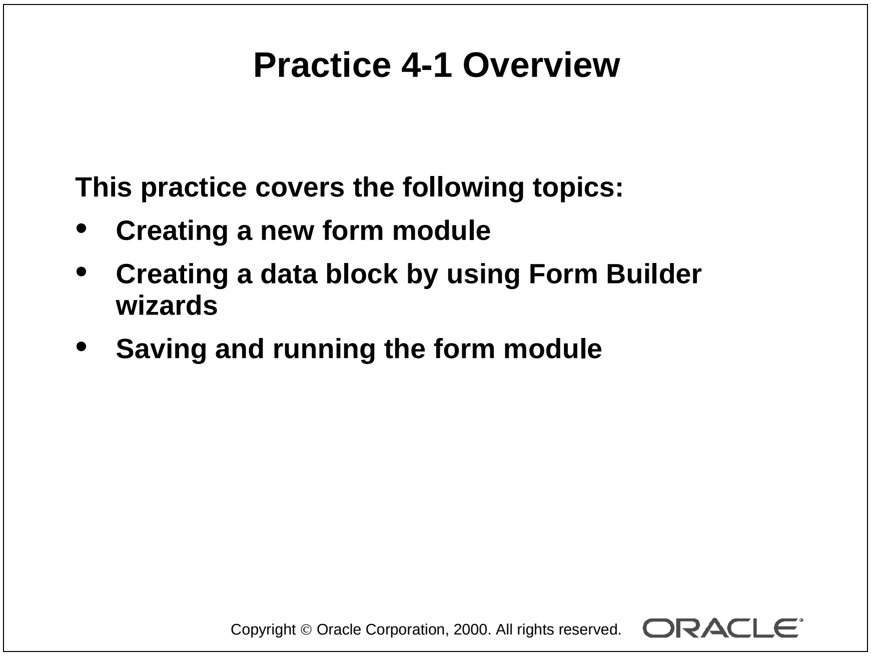
Copyright © Oracle Corporation, 2000. All rights reserved.

Module Type and Storage Format

When you create form modules, menu modules, and library documents in Form Builder they are stored in source files that have a binary format and are portable across platforms. The application files that your users run are also in a binary format; however, they are not portable across platforms.

Module/ Document	Extension	Storage Format	Portable
Form	`.fmb`	Form module binary	Yes
	`.fmx`	Form module executable; executable	No
	`.fmt`	Form module text	Yes
Menu	`.mmb`	Menu module binary	Yes
	`.mmx`	Menu module executable; executable	No
	`.mmt`	Menu module text	Yes
PL/SQL Library	`.pll`	PL/SQL Library document binary	Yes
	`.plx`	PL/SQL Library document executable (no source)	No
	`.pld`	PL/SQL Library document text	Yes
Object Library	`.olb`	Object Library module binary	Yes
	`.olt`	Object Library module text	Yes

Note: `.pll` is portable but requires recompilation, because it contains both source and compiled pcode.



Practice 4-1 Overview

This practice covers the following topics:

- **Creating a new form module**
- **Creating a data block by using Form Builder wizards**
- **Saving and running the form module**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 4-1, in Appendix A, “Practice Solutions.”

Instructor Note

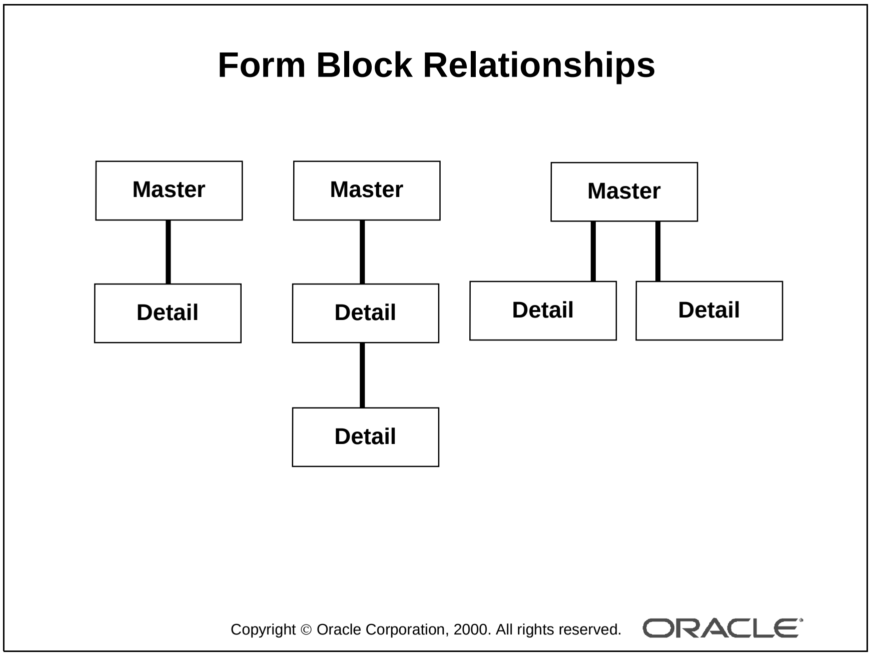
Advise the students not to spend too much time trying to get the layout of the CUSTOMER block to exactly match the screenshot provided in Appendix A, because the layout for the items in this block will be modified in a later lesson.

Practice 4-1 Overview

The practice session for this lesson is divided into Part 1 and Part 2.

In Part 1, you will create one new form module. You will create a single-block form that displays a single record.

- Create a new form module called CUSTOMERS. Create a new data block in this form by using the Form Builder wizards, and base it on the S_CUSTOMER table. Using the Layout Editor, reposition the items in this block to match the screenshot provided.
- Save and run the new form module on the Web.



Form Block Relationships

Master
Detail

Master
Detail
Detail

Master
Detail
Detail

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating Data Blocks with Relationships

A form module can contain one or more data blocks. Each data block can stand alone or be related to another data block.

Master-Detail Relationship

A master-detail relationship is an association between two data blocks that reflects a primary-foreign key relationship between the database tables on which the two data blocks are based. The master data block is based on the table with the primary key, and the detail data block is based on the table with the foreign key. A master-detail relationship equates to the one-to-many relationship in the entity relationship diagram.

A Detail Block Can Be a Master

You can create block relationships in which the detail of one master-detail link is the master for another link.

A Master Block Can Have More Details

You can create more than one detail block for a master block.

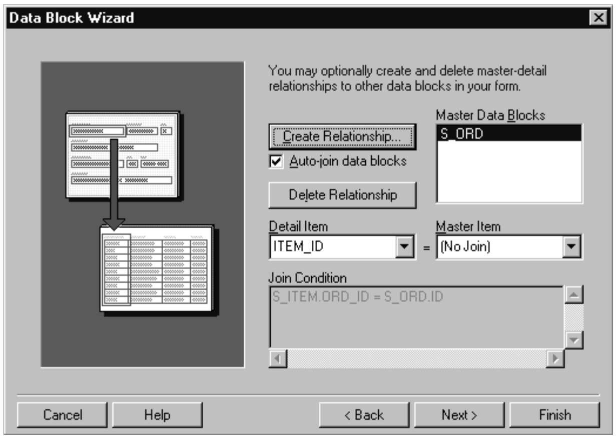
Note: The following are examples of the master-detail structure:

- Master-detail: Order-items
- Master-detail-detail: Customer-order-items
- Master-2*detail: Customer-order and customer-shipping contacts

Instructor Note

The Data Block Wizard also enables you to select an Oracle8 object when creating relationships.

Data Block Wizard: Master-Detail Page



Data Block Wizard

You may optionally create and delete master-detail relationships to other data blocks in your form.

Master Data Blocks: S_ORD

☑ Auto-join data blocks

Detail Item: ITEM_ID = Master Item: (No Join)

Join Condition: S_ITEM.ORD_ID = S_ORD.ID

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Create a second data block, using the Form Builder wizards. Base the block on the S_ITEM table. In the Data Block Wizard, create a master-detail relationship between the new data block and the S_ORD data block.

Creating a Master-Detail Form Module with the Data Block Wizard

You can build a master-detail form module either by creating a relation between a master and detail block explicitly, or by using the Data Block Wizard.

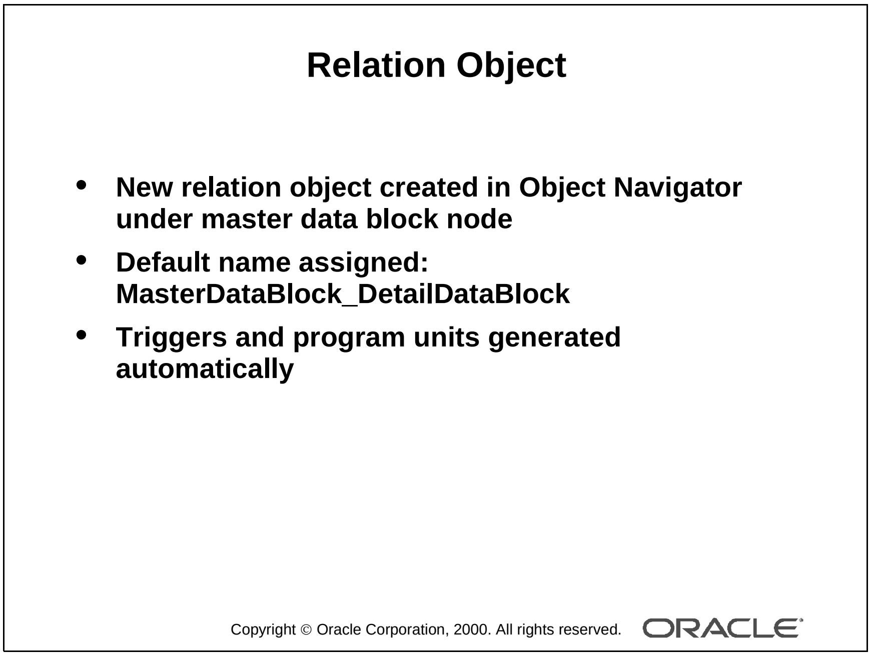
1. Create the master block as described earlier in this lesson.
2. Invoke the Data Block Wizard in the Object Navigator.
3. Follow the same steps as before to create a new data block in the Data Block Wizard until you come to the Master-Detail page. On this page, select the “Auto-join data blocks” check box and click Create Relationship.

   Note: If the “Auto-join data blocks” check box is clear, the Data Block dialog is displayed with a list of *all data blocks* in the form without any foreign key constraint names.
4. Select a master data block in the Data Block dialog and click OK. The wizard automatically creates the join condition between the detail and master data blocks in the Join Condition field and displays the name of the master data block in the Master Data Blocks field.

   Note: If the “Auto-join data blocks” check box is clear, the wizard does not automatically create the join condition between the detail and master data blocks. You must use the Detail Item and Master Item pop-up lists to create a join condition manually.
5. Click Next and finish the Data Block Wizard steps. Go through the Layout Wizard steps as described earlier in this lesson to finish creating and laying out the detail data block.

Note: The master data block must exist in the form module before you create the detail block.

You can also create a relation by invoking the Data Block Wizard in reentrant mode.



Relation Object

- **New relation object created in Object Navigator under master data block node**
- **Default name assigned: MasterDataBlock_DetailDataBlock**
- **Triggers and program units generated automatically**

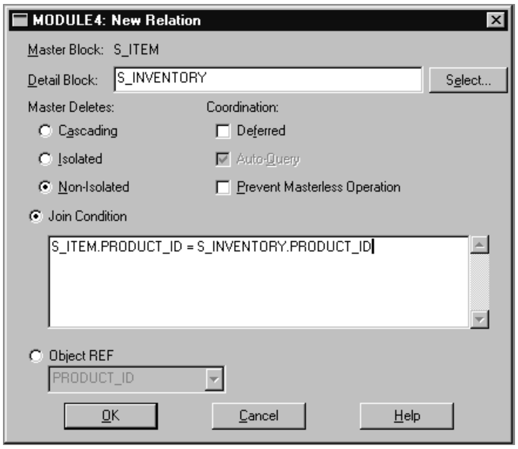
Copyright © Oracle Corporation, 2000. All rights reserved.

New Relation

Once you create a master-detail form module, the Data Block Wizard automatically creates a form object that handles the relationship between two associated data blocks. This object is called a *relation*. The following tasks occur automatically:

- The new relation object is created under the master data block node in the Object Navigator with default properties.
- The relation is given the following default name: MasterDataBlock_DetailDataBlock, for example S_ORD_S_ITEM
- Triggers and program units are generated to maintain coordination between the two data blocks.

Creating a Relation Manually



MODULE4: New Relation

Master Block: S_ITEM

Detail Block: S_INVENTORY

Master Deletes:

- ○ Cascading
- ○ Isolated
- ● Non-Isolated

Coordination:

- □ Deferred
- ☑ Auto-Query
- □ Prevent Masterless Operation

● Join Condition

```
S_ITEM.PRODUCT_ID = S_INVENTORY.PRODUCT_ID
```

○ Object REF

PRODUCT_ID

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Relation Manually

What Is a Relation?

A *relation* is a Form Builder object that handles the relationship between two associated blocks.

You can create a relation either:

- Implicitly with a master-detail form module
- Explicitly in the Object Navigator

Implicit Relations

When you create a master-detail form module, a relation is automatically created. This relation is named masterblock_detailblock, for example, S_ORD_S_ITEM.

Explicit Relations

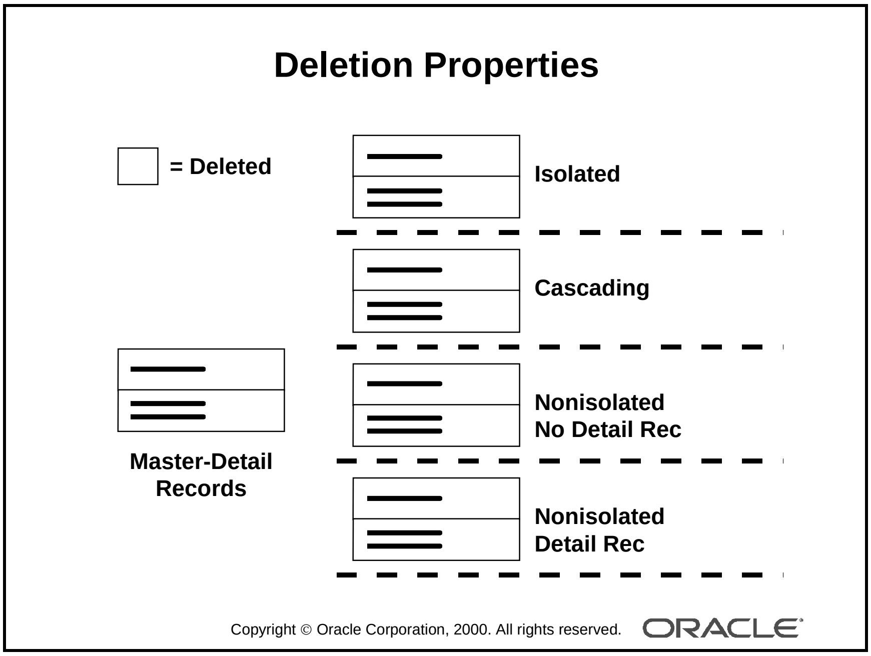
If a relation is not established when default blocks are created, you can create your own by setting the properties in the New Relation dialog box. Like implicitly created relations, PL/SQL program units and triggers are created automatically when you explicitly create a relation.

How to Create a Relation Explicitly

1. Select the master block entry in the Object Navigator.
2. Click the Create icon.

   The New Relation window is displayed.
3. Specify the name of the detail block.
4. Choose your master delete property.
5. Choose your coordination property.
6. Specify the join condition.
7. Click OK.

The new relation, new triggers, and new program units are highlighted in the Object Navigator.



Deletion Properties

= Deleted

Isolated

Cascading

Nonisolated
No Detail Rec

Nonisolated
Detail Rec

Master-Detail
Records

Copyright © Oracle Corporation, 2000. All rights reserved.

Modifying a Relation

You can alter the relation properties to affect the way deletes and block coordination are handled.

Master Deletes

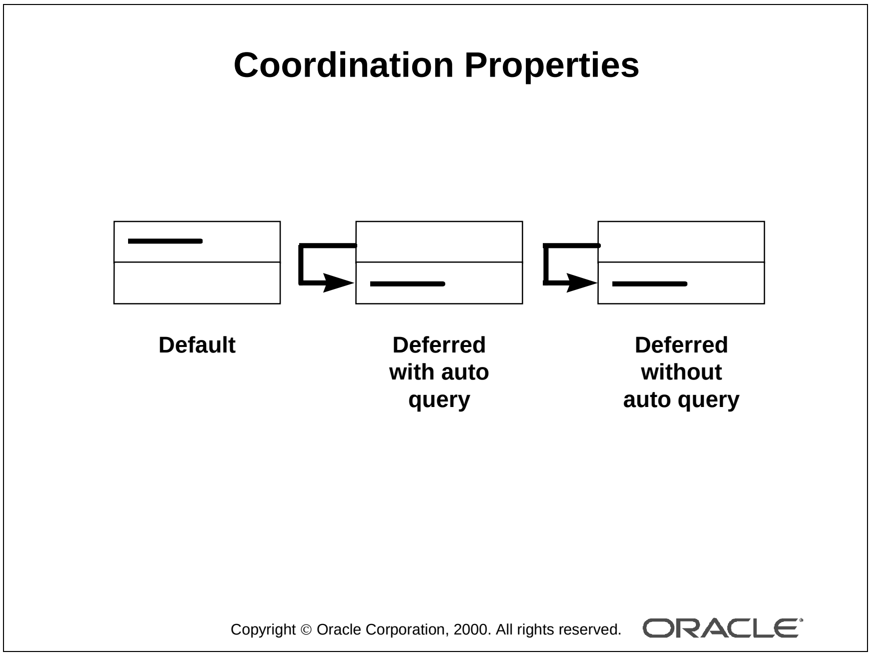
You can prevent, propagate, or isolate deletion of a record in a master block when corresponding records exist in the detail block by setting the Delete Record Behavior property. For example, you can delete all corresponding line items when an order is deleted.

Property	**Use**
Non-Isolated	Prevents the deletion of the master record when the detail records exist
Cascading	Deletes the detail records when a master record is deleted
Isolated	Deletes only the master record

Note: Although deleting with the cascading property may remove many detail records, the commit message shows only the number of records deleted from the master block.

What Happens When You Modify a Relation?

- Changing the Delete Record Behavior property from the default of Non-Isolated to Cascading replaces the On-Check-Delete-Master trigger with the Pre-Delete trigger.
- Changing the Delete Record Behavior property from the default of Non-Isolated to Isolated results in the removal of the On-Check-Delete-Master trigger.



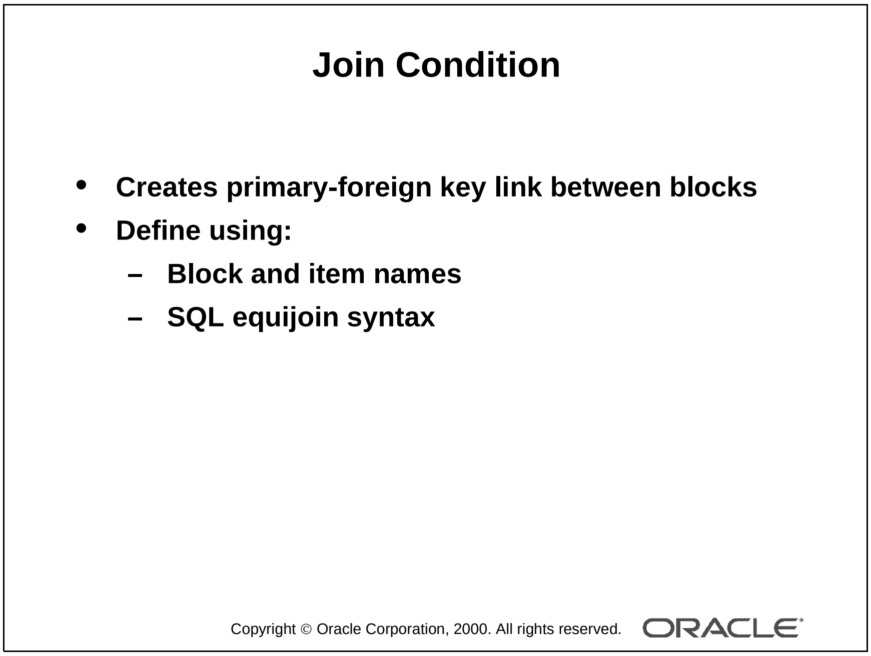
Coordination Properties

Default

Deferred with auto query

Deferred without auto query

Copyright © Oracle Corporation, 2000. All rights reserved.



Join Condition

- Creates primary-foreign key link between blocks
- Define using:
  - Block and item names
  - SQL equijoin syntax

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Explicitly add a relation between the Department and Employee blocks of `explrela.fmb`. Set the deletion property to Cascading and the coordination property to Deferred with Auto-Query.

Coordination

You can control how the detail records are displayed when a master block is queried by setting the coordination property. For example, you can defer querying the line items for an order until the operator navigates to the item block.

Coordination Property	**Use**
Default	Forces coordination of blocks to occur whenever the master record is changed by a user or a trigger
Deferred with Auto Query	Postpones potentially expensive detail query processing until the cursor visits the related blocks
Deferred Without Auto Query	Allows entry of additional query criteria in the detail block prior to querying
Prevent Masterless Operations	Ensures that the detail block cannot be queried or used to insert records when a master record is not displayed

Note: Setting the Deferred property to Yes enables the Auto Query check box.

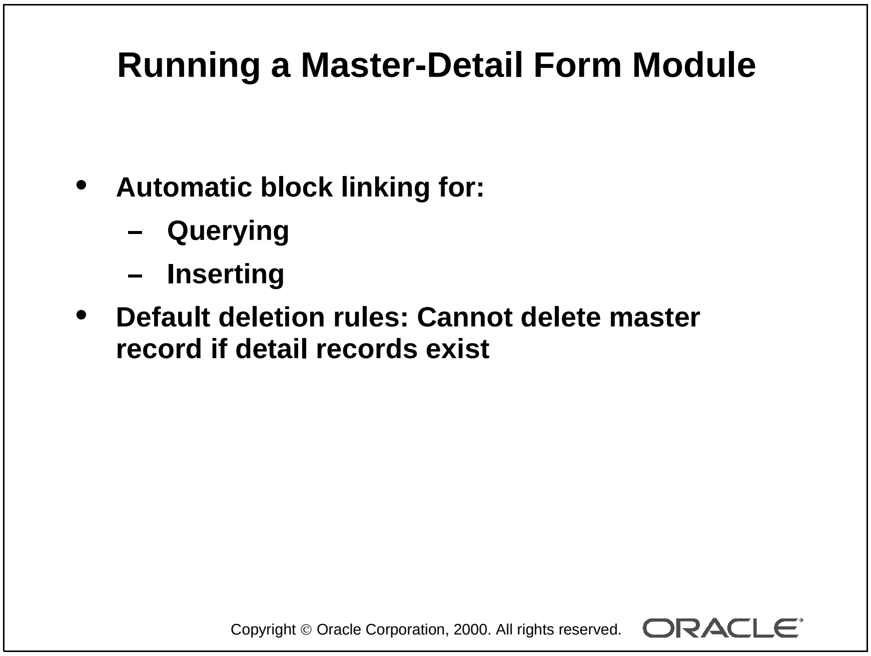
Join Condition

Use to:

- Create links between blocks using SQL
- Alter links between blocks using SQL

Define using:

- Usual SQL equi-join condition syntax
- Block names instead of the base table names
- Item names that exist in the form module instead of base table column names



Running a Master-Detail Form Module

- Automatic block linking for:
  - Querying
  - Inserting
- Default deletion rules: Cannot delete master record if detail records exist

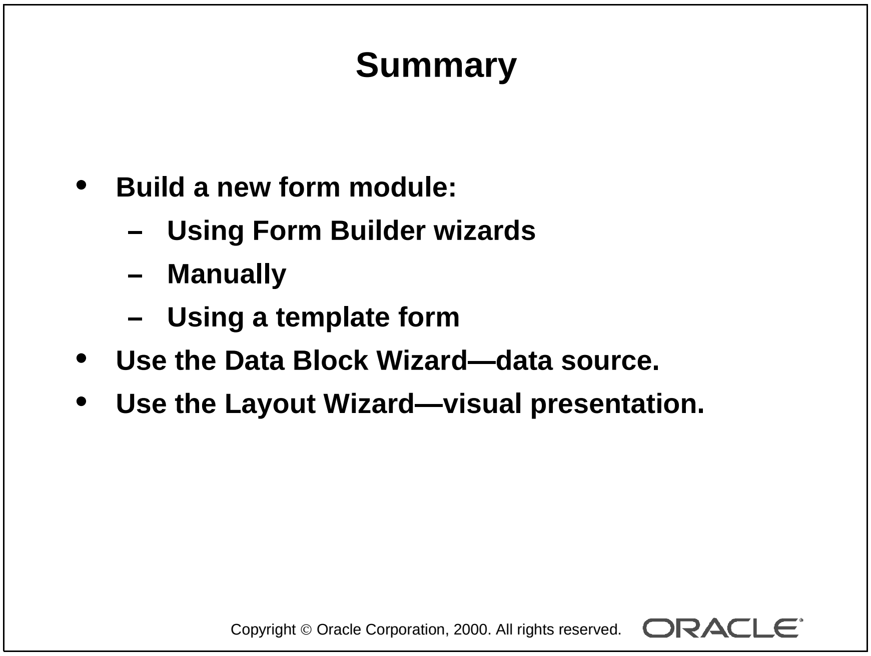
Copyright © Oracle Corporation, 2000. All rights reserved.

Running a Master-Detail Form Module

When you run your master-detail form module you will find that:

- Querying the master data block immediately retrieves corresponding detail records.
- Deleting a master record is prevented if detail records exist.

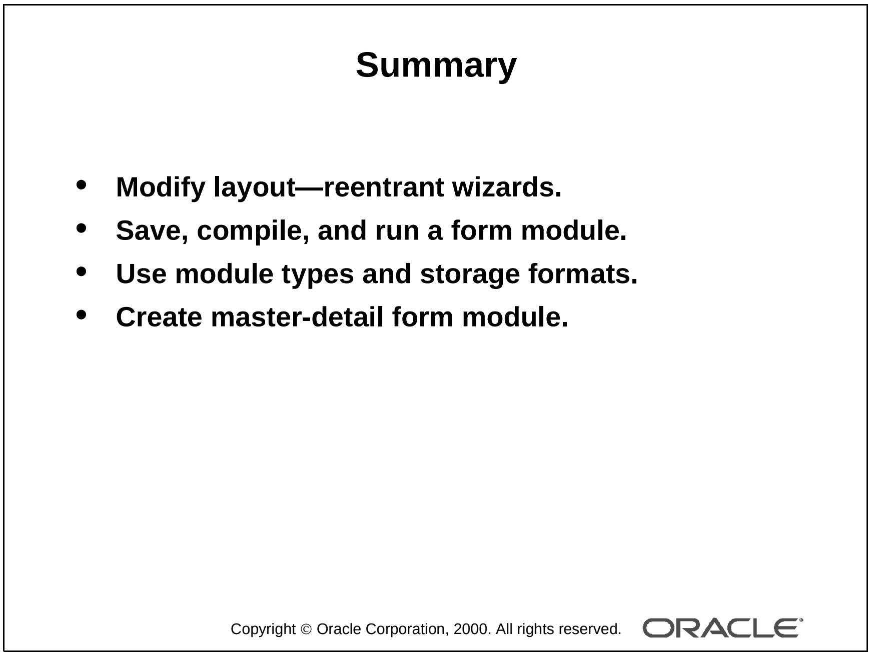
  Note: You can change the above behavior by modifying the relation object properties.
- Inserting a detail record automatically associates it with the currently displayed master.



Summary

- **Build a new form module:**
  - **Using Form Builder wizards**
  - **Manually**
  - **Using a template form**
- **Use the Data Block Wizard—data source.**
- **Use the Layout Wizard—visual presentation.**

Copyright © Oracle Corporation, 2000. All rights reserved.



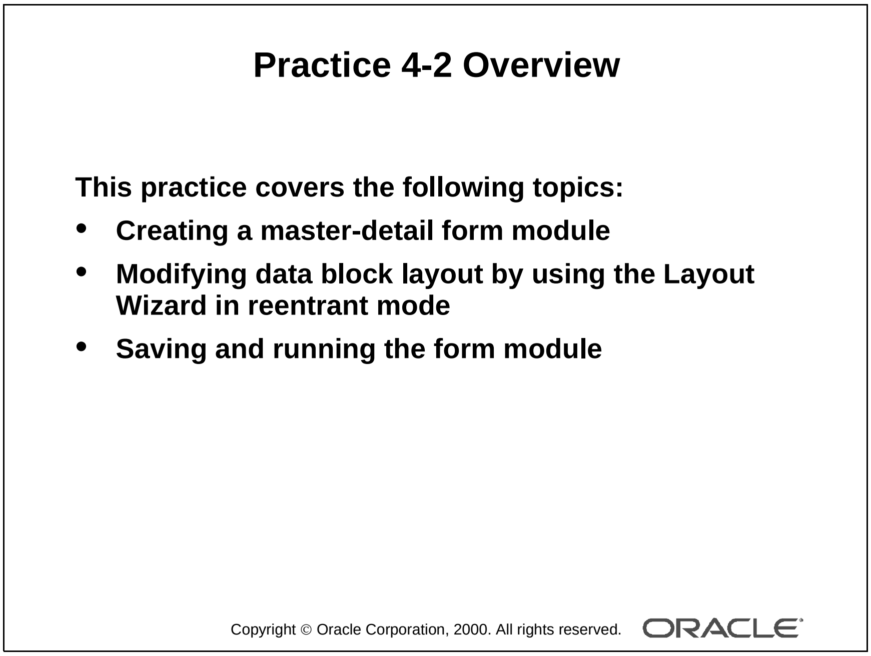
Summary

- **Modify layout—reentrant wizards.**
- **Save, compile, and run a form module.**
- **Use module types and storage formats.**
- **Create master-detail form module.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

- Building a new form module by using the following methods:
  - Form Builder wizards
  - Manually
  - Template form
- Using the Data Block Wizard to create a new data block with its associated data sources quickly and easily
- Using the Layout Wizard to quickly lay out the new data block contents for user interaction
- Modifying the data block layout:
  - Using reentrant wizards
  - Changing frame properties
- Saving the form module to preserve its definition; compiling it to get an executable file; running the form module to test it
- Using several module types and storage formats that are available for form modules, menu modules, PL/SQL Library documents, and Object Library modules
- Creating data blocks with a master-detail relationship



Practice 4-2 Overview

This practice covers the following topics:

- **Creating a master-detail form module**
- **Modifying data block layout by using the Layout Wizard in reentrant mode**
- **Saving and running the form module**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 4-2, in Appendix A, “Practice Solutions.”

Practice 4-2 Overview

In Part 2, you will create a new form module that displays master-detail information.

- Create a master-detail form module called ORDERS. Create a master block based on the S_ORD table and a detail block based on the S_ITEM table. Create a third data block that is not related to any other block in the form module. Base this block on the S_INVENTORY table, and manually create a relation with the block based on the item table. Use the Form Builder wizards to create all three data blocks.
- Invoke the Layout Wizard in reentrant mode, and change the layout of the S_ITEM and S_INVENTORY data blocks.
- Save and run the new form module on the Web.

Practice 4-1

1 Create a new form module.

Create a new single block by using the Data Block Wizard.

Base it on the S_CUSTOMER table and include all columns.

Display the S_CUSTOMER block with a form style layout on a new content canvas called CV_CUSTOMER and show just one record at a time. Set the frame title to Customers.

Hint: Manually rename the canvas in the Object Navigator.

2 Save the new module to a file called CUSTG*XX*, where *XX* is the group number that your instructor has assigned to you.

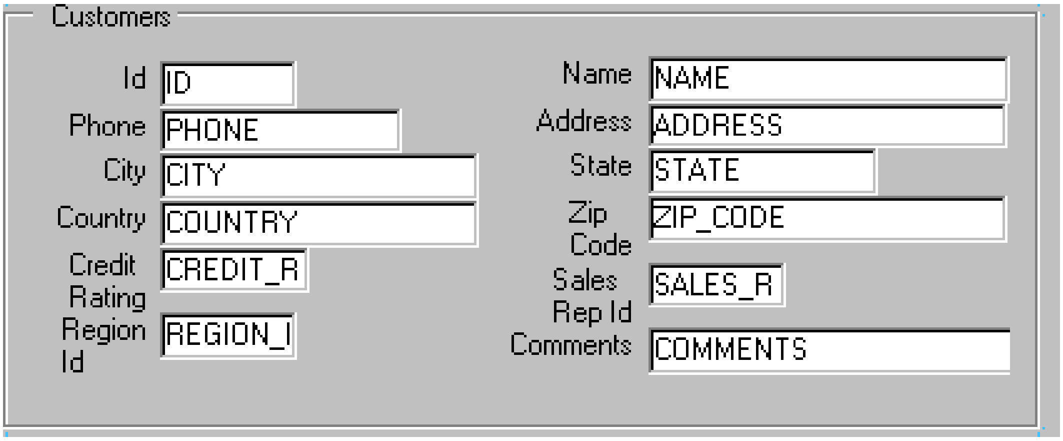
Run your form module and execute a query.

Navigate through the fields. Exit run time and return to Form Builder.

3 Change the form module name in the Object Navigator to CUSTOMERS.

4 In the Layout Editor, reposition the items, so that the canvas resembles the screenshot below.

Hint: First resize the canvas and the frame.



Customers

Id ID

Phone PHONE

City CITY

Country COUNTRY

Credit Rating CREDIT_R

Region Id REGION_I

Name NAME

Address ADDRESS

State STATE

Zip Code ZIP_CODE

Sales Rep Id SALES_R

Comments COMMENTS

5 Save and compile the form module.

Deploy your form module on the Web and execute a query.

Practice 4-2

1 Create a new form module.

Create a new block by using the Data Block Wizard.

Base it on the S_ORD table and include all columns except TOTAL.

Display the S_ORD block on a new content canvas called CV_ORDER and show just one record at a time. Use a form style layout. Set the frame title to Orders.

Hint: Manually rename the canvas in the Object Navigator.

2 Create a new block by using the Data Block Wizard.

Base the block on the S_ITEM table and include all columns.

Create a relationship and select the master block as S_ORD.

Display all items except ORD_ID on the CV_ORDER canvas.

Display six records in this detail block on the same canvas as the master block.

Use a tabular style layout and include a scroll bar.

Change the order of the blocks in the Object Navigator, moving the S_ITEM block after the S_ORD block. Set the frame title to Items.

3 Save the new module to a file called `ORDG`*`XX`*, where *XX* is the group number that your instructor has assigned to you.

4 Create a new block based on S_INVENTORY (do not create any relationships with other blocks at this time) to display on a different canvas.

Base it on the S_INVENTORY table; exclude the OUT_OF_STOCK_EXPLANATION column from the definition.

Display four records in this block and ensure that they display on a new content canvas called CV_INVENTORY.

Use a tabular style layout, and include a scroll bar.

In the Object Navigator move the S_INVENTORY block after the S_ITEM block. Set the frame title to Stock.

Do not create any relationships between blocks at this time.

5 Create a relation called S_Item_S_Inventory explicitly between the S_Item and S_Inventory blocks.

Ensure that line item records can be deleted independently of any related inventory.

Set the coordination so that the Inventory block is not queried until you explicitly execute a query.

6 On the S_ITEM block change the prompt for the Quantity Shipped item to Shipped by using the reentrant Layout Wizard. First select the relevant frame in the Layout editor, then use Layout Wizard.

7 In the S_INVENTORY data block, change the prompt for Amount in Stock to In Stock by using the Layout Wizard.

8 Save and compile your form module.

Run your form module on the Web.

Execute a query.

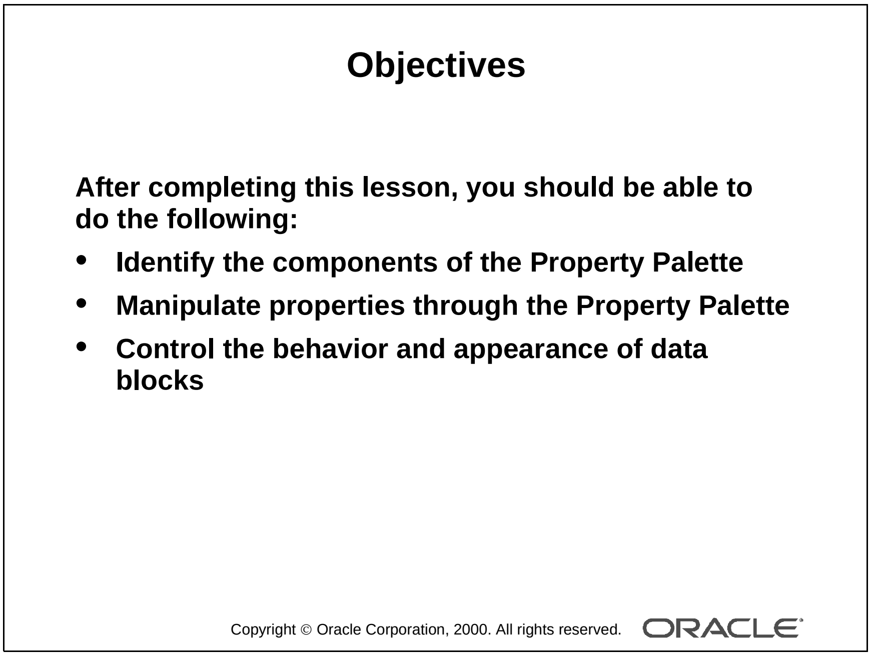
Navigate through the blocks so that you see the S_INVENTORY block.

Exit run time, close Netscape Communicator, and return to Form Builder.

9 Change the form module name in the Object Navigator to ORDERS and save.

5

Working with Data Blocks and Frames

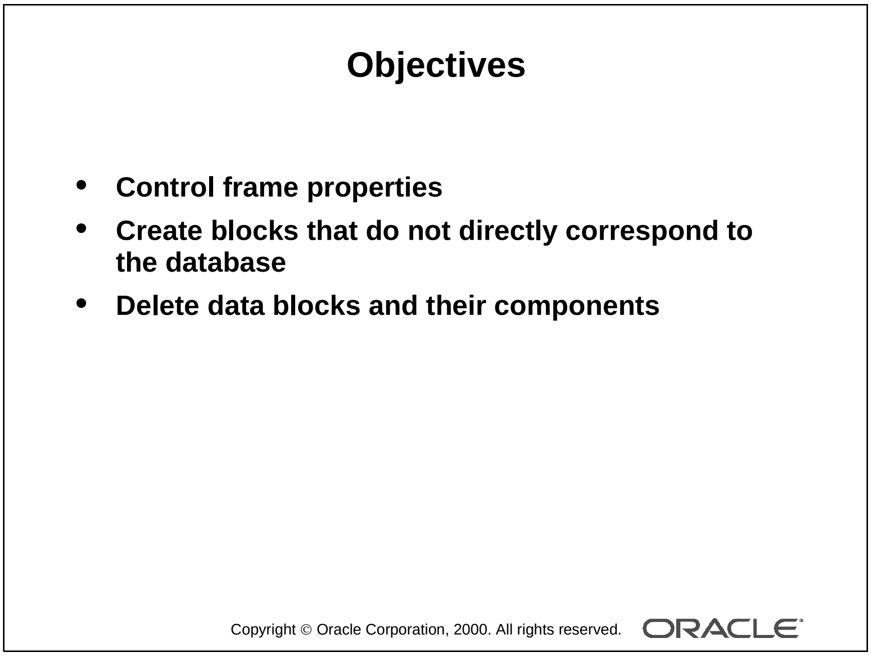


Objectives

After completing this lesson, you should be able to do the following:

- **Identify the components of the Property Palette**
- **Manipulate properties through the Property Palette**
- **Control the behavior and appearance of data blocks**

Copyright © Oracle Corporation, 2000. All rights reserved.



Objectives

- **Control frame properties**
- **Create blocks that do not directly correspond to the database**
- **Delete data blocks and their components**

Copyright © Oracle Corporation, 2000. All rights reserved.

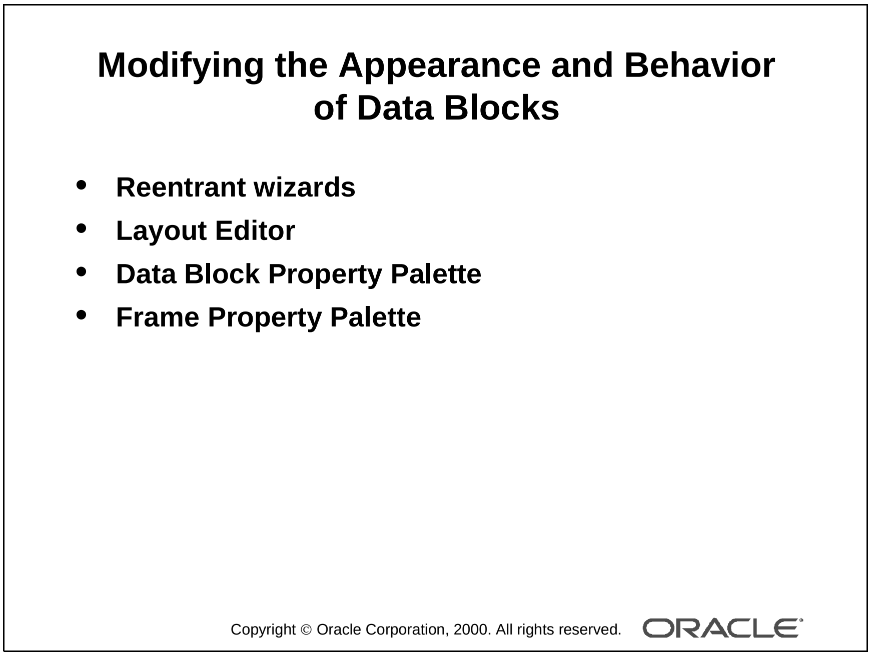
Schedule

Topic	**Timing**
Lecture	60 minutes
Practice	40 minutes
Total	100 minutes

Introduction

Overview

In this lesson you will learn how to customize existing data blocks and modify frames. You will also learn how to include blocks that are not associated with the database.



Modifying the Appearance and Behavior of Data Blocks

- Reentrant wizards
- Layout Editor
- Data Block Property Palette
- Frame Property Palette

Copyright © Oracle Corporation, 2000. All rights reserved.

Managing Object Properties

Modifying the Appearance and Behavior of Data Blocks

You can modify the appearance and behavior of a data block after it has been created. To do this use one of the following methods:

- Reentrant wizards: Reenter the Data Block Wizard or the Layout Wizard as described in the previous lesson to modify the data source and visual presentation of objects within the data block.
- Layout Editor: Invoke the Layout Editor and make your modifications manually.
- Data Block Property Palette: Open the Data Block Property Palette and change individual property values to modify the behavior of the data block at run time.
- Frame Property Palette: Open the associated Frame Property Palette and change individual property values to modify the arrangement of items within the data block.

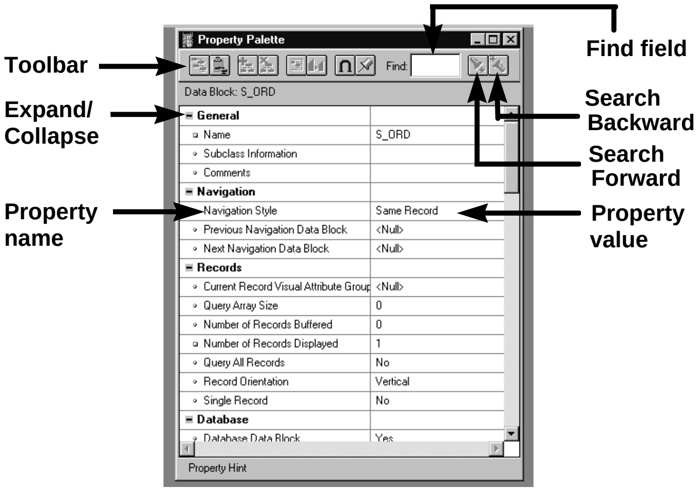
Displaying the Property Palette

Use one of several methods:

- **Select Tools—>Property Palette.**
- **Double-click the object icon in the Object Navigator.**
- **Double-click the object in Layout Editor.**
- **Click the right mouse button.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Property Palette Features



Toolbar

Expand/ Collapse

Property name

Find field

Search Backward

Search Forward

Property value

Property Palette

Find:

Data Block: S_ORD

General	
Name	S_ORD
Subclass Information	
Comments	
Navigation	
Navigation Style	Same Record
Previous Navigation Data Block	<Null>
Next Navigation Data Block	<Null>
Records	
Current Record Visual Attribute Group	<Null>
Query Array Size	0
Number of Records Buffered	0
Number of Records Displayed	1
Query All Records	No
Record Orientation	Vertical
Single Record	No
Database	
Database Data Block	Yes

Property Hint

Copyright © Oracle Corporation, 2000. All rights reserved.

Property Palette

Every object in a form module, as well as the form module itself, has properties that dictate the object’s behavior. When an object is first created, it is automatically assigned several property values by default. You can change these property values in the Property Palette.

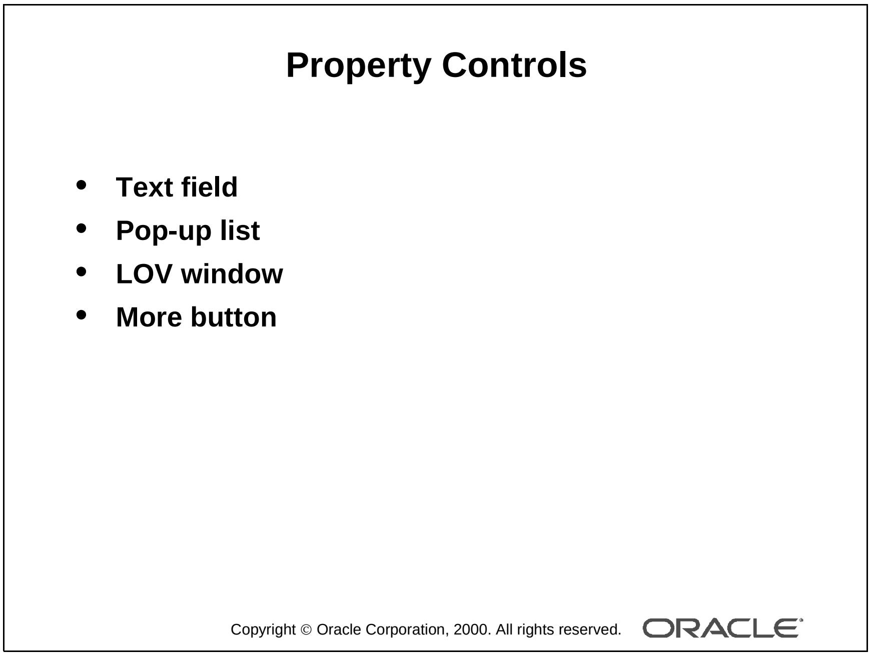
Displaying the Property Palette

To display the Property Palette of an object, use any of the following methods:

- Select the object in the Object Navigator and then select Tools—>Property Palette from the menu system.
- Double-click the object icon for the object in the Object Navigator (except for code objects and canvases).
- Double-click an item in the Layout Editor.
- Select the object in the editor or the Object Navigator, and then click the right mouse button. From the pop-up menu, select the Property Palette option.

Property Palette Features

Feature	Description
Property list	The property list displays a two-column list of property names and property values. Properties are grouped under functional headings or nodes. You can expand or collapse a node by using the plus and minus icons beside the node name.
Find field	The Find field enables you to quickly locate the name of a particular property. The Search Forward and Search Backward buttons enhance your search.
Toolbar	The toolbar consists of a series of buttons that provide quick access to commands.



Property Controls

- **Text field**
- **Pop-up list**
- **LOV window**
- **More button**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

Once you activate the Property Palette for an object, its window remains open. The window automatically displays the properties of each object you visit in the Layout Editor or Object Navigator. This is because, by default, the list of properties in the Property Palette is synchronized whenever you select an object.

You can turn the synchronization on or off for a specific palette by clicking Freeze/ Unfreeze in that Property Palette toolbar.

Using the Property Palette

Each form object has various types of properties. Properties are manipulated differently, depending on the property type. Here is a summary of controls used in the Property Palette:

Property Control	Description
Text field	This is displayed when the current property can be set by entering a text value. For longer text values, an iconic button also appears, enabling you to open a text editor.
Poplist	This occurs where the property is Yes or No, or where a fixed set of values are allowed. Click the down arrow to open the list and select a value. Alternatively, double-click the property name to cycle through the values.
LOV window	LOVs occur where a potentially large list of possible values is available. Click the iconic button in the property value column to invoke an LOV.
More button	Use this when more complex settings are needed. Click the More button to open the extra dialog.

Property Palette Icons

Each property in a Property Palette has an icon to its left. Here is a summary of these icons and their description:

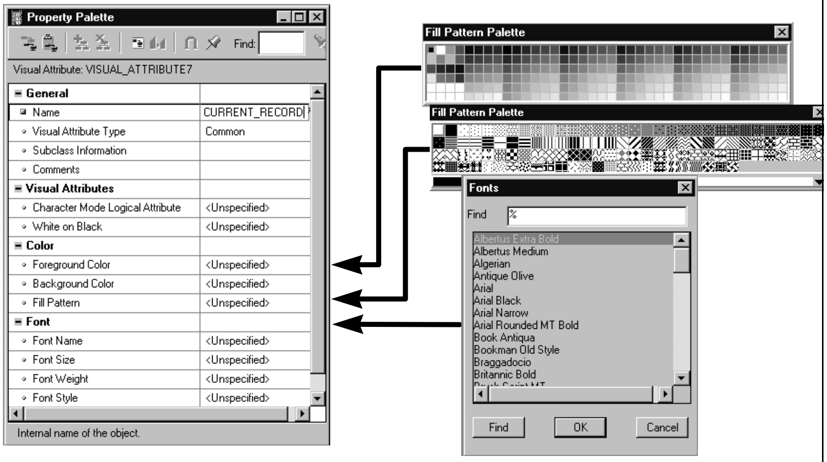
Icon	Description
Circle	Specifies that the property value is the default value
Square	Specifies that the property value has been changed from the default
Arrow	Specifies that the property value is inherited
Arrow with a cross	Specifies that the property value was inherited but has been overridden

Visual Attributes

- **Visual attributes:**
  - **Are font, color, and pattern properties**
  - **Can be set for form and menu objects**
- **A visual attribute is a form object with font, color, and pattern properties.**
- **Set the Visual Attribute Group property to the visual attribute object.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Font, Pattern, and Color Picker



Visual Attribute: VISUAL_ATTRIBUTE7

General	
Name	CURRENT_RECORD
Visual Attribute Type	Common
Subclass Information	
Comments	
Visual Attributes	
Character Mode Logical Attribute	<Unspecified>
White on Black	<Unspecified>
Color	
Foreground Color	<Unspecified>
Background Color	<Unspecified>
Fill Pattern	<Unspecified>
Font	
Font Name	<Unspecified>
Font Size	<Unspecified>
Font Weight	<Unspecified>
Font Style	<Unspecified>

Internal name of the object.

Find %

Albertus Extra Bold
Albertus Medium
Algerian
Antique Olive
Arial
Arial Black
Arial Narrow
Arial Rounded MT Bold
Book Antiqua
Bookman Old Style
Braggadocio
Britannic Bold

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Create a visual attribute to be used later in this lesson. Use the Color, Pattern, and Font pickers. Use the `ordwk04.fmb` file.

Visual Attributes

Visual attributes are the font, color, and pattern properties that you set for form and menu objects.

A visual attribute is another object that you can create in the Object Navigator with properties such as font, color, and pattern combinations.

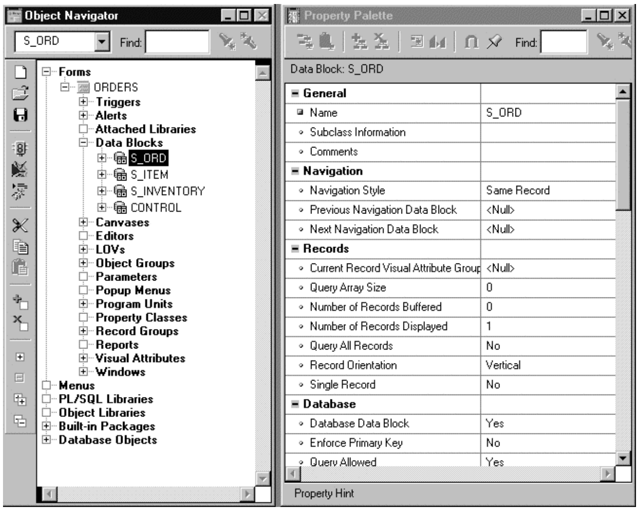
When creating visual attributes, you can use the Font Picker and Color Picker to select the font and color. When changing a font from the Property Palette, click the Font group itself to invoke the Font Picker.

Every interface object in a forms application has a property called Visual Attribute Group, which determines how the individual visual attribute settings of an object are derived. The Visual Attribute Group property can be set to Default, NULL, or the name of a named visual attribute object.

Partial Visual Attributes

Partial visual attributes are just like visual attributes, except that you need to set only the properties that you want to be inherited by the objects that use them. This means that you can apply a visual attribute that changes the font color without having to set the font name.

Data Block Properties



Data Block: S_ORD

General	
Name	S_ORD
Subclass Information	
Comments	
Navigation	
Navigation Style	Same Record
Previous Navigation Data Block	<Null>
Next Navigation Data Block	<Null>
Records	
Current Record Visual Attribute Group	<Null>
Query Array Size	0
Number of Records Buffered	0
Number of Records Displayed	1
Query All Records	No
Record Orientation	Vertical
Single Record	No
Database	
Database Data Block	Yes
Enforce Primary Key	No
Query Allowed	Yes

Copyright © Oracle Corporation, 2000. All rights reserved.

Data Block Property Groups

- **General**
- **Navigation**
- **Records**
- **Database**
- **Advanced Database**
- **Scrollbar**
- **Font and Color**
- **Character Mode**
- **International**

Copyright © Oracle Corporation, 2000. All rights reserved.

Data Block Properties

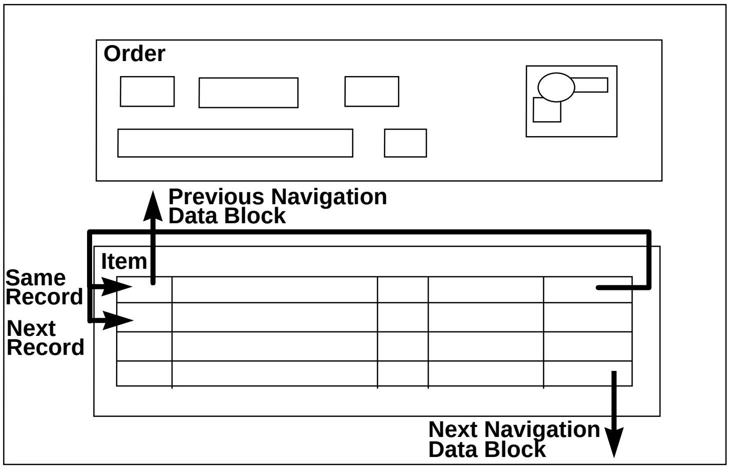
Each data block has several properties. These properties are divided into the following groups:

- General
- Navigation
- Records
- Database
- Advanced Database
- Scrollbar
- Font and Color
- Character Mode
- International

Instructor Note

The properties of several objects have been renamed from previous versions. To get a detailed listing of the renamed properties, refer to the “Renamed properties” topic in the online documentation. In the Help Topics: Form Builder Help window, select the Contents tab, and expand the relevant topics until you get to the “Renamed properties” topic.

Navigation Properties



Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Use the `ordwk04.fmb` file to show the run-time effect of changing the Navigation Style property on the S_ITEM block.

Controlling the Behavior of Data Blocks

Setting Navigation Properties

Navigation Style Normally, when you navigate beyond the last item in a record, Form Builder returns you to the beginning of the same record. With this property you can change the location of the cursor.

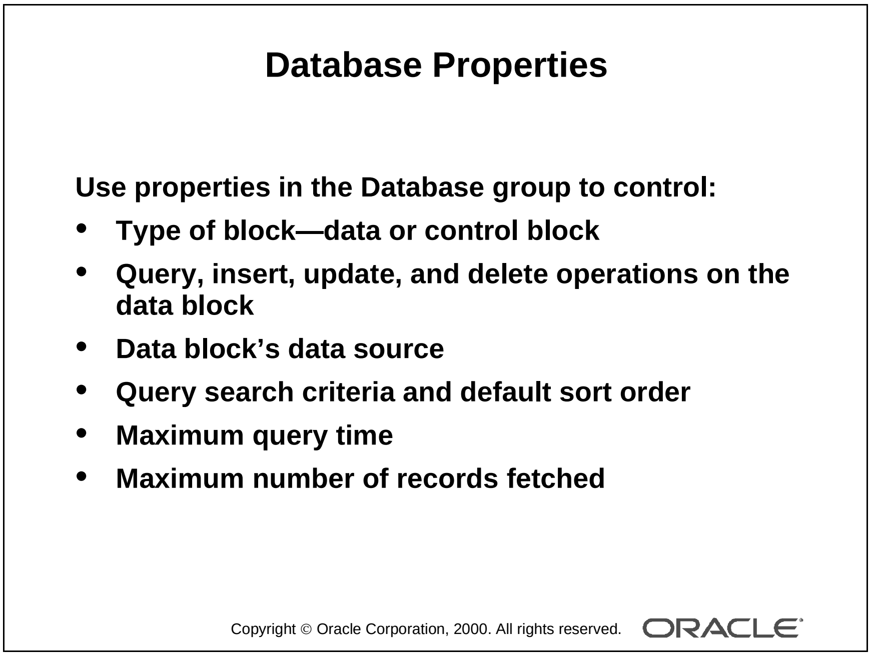
The following settings are valid for the Navigation Style property:

- Same Record (default)
- Change Record
- Change Data Block

Note: If you want the cursor to move to the next record when you reach the end of the current record, set the Navigation Style property for the block to Change Record.

Previous/Next Navigation Data Block Each data block in a form module has a sequenced position in the data block object hierarchy. Normally, when you perform an operation to move to the previous or next data block at run time, Form Builder moves control to the previous or next adjacent data block in sequence. These properties enable you to name the previous or next data block to which to navigate from the current one.

.



Database Properties

Use properties in the Database group to control:

- **Type of block—data or control block**
- **Query, insert, update, and delete operations on the data block**
- **Data block’s data source**
- **Query search criteria and default sort order**
- **Maximum query time**
- **Maximum number of records fetched**

Copyright © Oracle Corporation, 2000. All rights reserved.

Setting Database Properties

Database Data Block This property is set to Yes if the data block is based on a database object and No if it is a control block.

Enforce Primary Key This controls whether Form Builder checks that records are unique before they are inserted or updated in the base table, in order to avoid committing duplicate rows in the database. A value of Yes means that the form checks that inserted or updated records in the client-side record group are unique before an attempt is made to commit possible duplicate rows.

Query/Insert/Update/Delete Allowed These properties control whether the associated operations can be performed on the data block records.

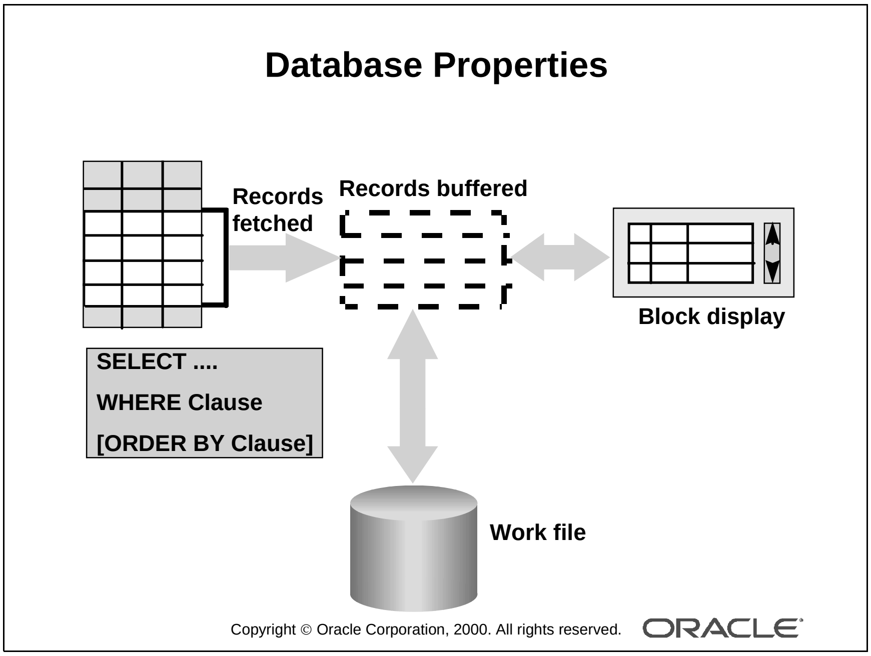
Query Data Source Type This property specifies the type of the query data source for the data block. Possible values for this property are None, Table, Procedure, Transactional Triggers, or FROM clause query.

Query Data Source Name This property specifies the name of the query data source for the data block. This property is used only if the type of the query data source is Table, FROM clause query, or Procedure.

Query Data Source Columns This property specifies, in a dialog box, the name and data type of the columns associated with the query data source. This property is used only if the type of the query data source is Table, FROM clause query, or Procedure.

WHERE Clause This property specifies a SQL condition that is attached to every default SELECT statement associated with the data block through implicit SQL. This clause is automatically appended (ANDed) with any conditions supplied by the operator in Enter Query mode.

Use this property to define general restrictions on the rows this data block may fetch.



Database Properties

Records fetched

Records buffered

Block display

SELECT ....

WHERE Clause

[ORDER BY Clause]

Work file

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Instructor Note

When Update Changed Columns Only is set to No, Form Builder can reuse the same SQL statement for multiple updates without having to reparse each time in the system global area (SGA). Changing this property value to Yes can degrade performance because the update statement must be reparsed each time.

Setting Database Properties (continued)

ORDER BY Clause This clause defines a default order for records displayed from a query. The operator can alter this order by using the Query Where dialog box at run time.

Optimizer Hint This property specifies a hint string that Form Builder passes to the Optimizer when constructing implicit SQL on the data block. Using the Optimizer can improve the performance of database transactions.

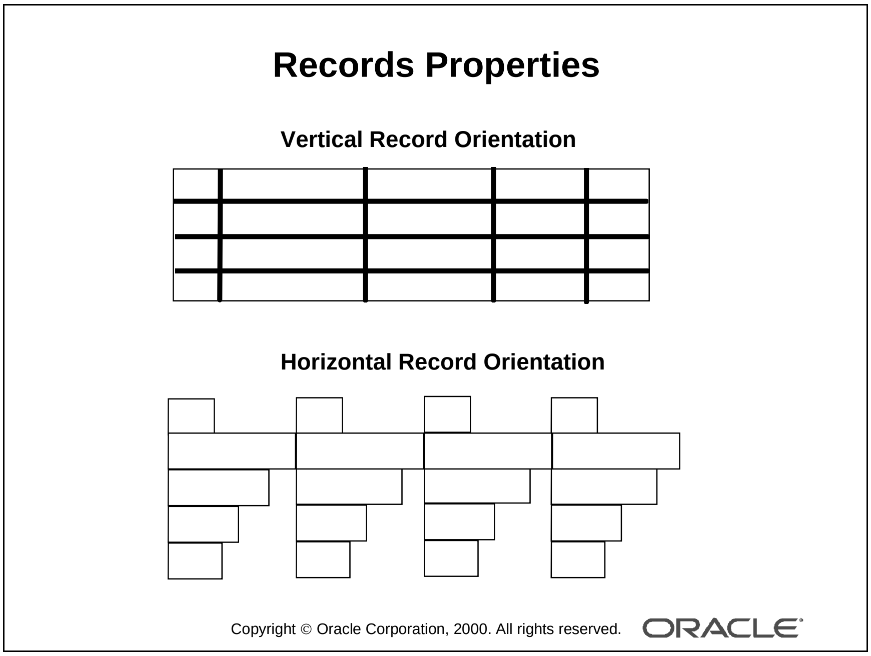
Locking Mode/Key Mode These properties control how Form Builder handles records and transactions when the data block is primarily associated with non-Oracle data sources. The default settings are usually appropriate for data blocks connected with an Oracle database.

Update Changed Columns Only When this property is set to Yes, only those items updated by the operator are written to their corresponding database columns. If the operator commonly updates or inserts records with only one or two columns, this can save network traffic. By default, this property value is set to No, so that all columns are included in the default UPDATE statement.

Enforce Column Security When this property is set to Yes, items in the data block can be updated only if the current user has permission to update the corresponding database columns.

Maximum Query Time This property provides the option to abort a query when the elapsed time of the query exceeds the value of this property. This property is useful when the Query All Records property is set to Yes.

Maximum Records Fetched This property provides the option to abort a query when the number of records fetched exceeds the value of this property. This property is useful when the Query All Records property is set to Yes.

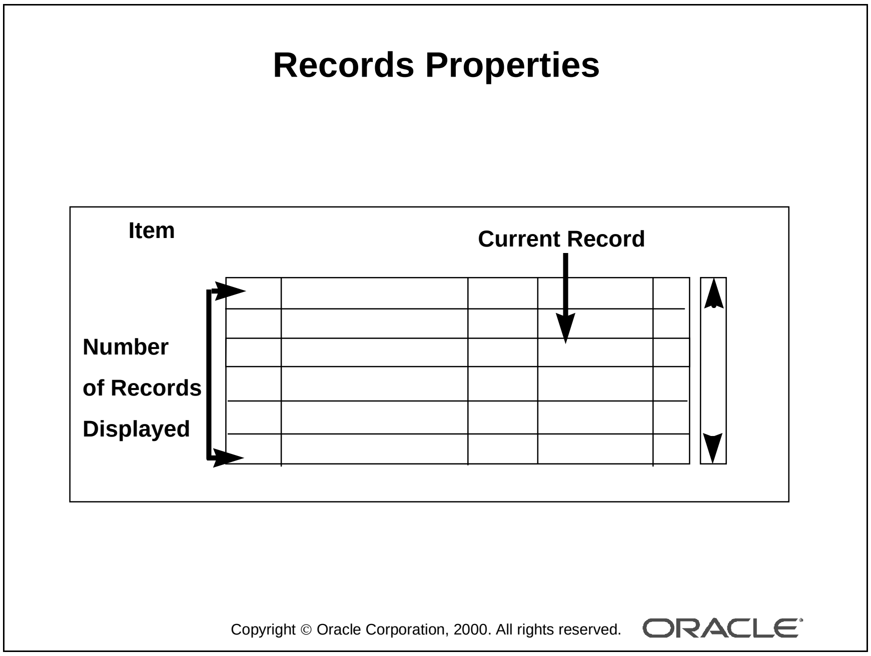


Records Properties

Vertical Record Orientation

Horizontal Record Orientation

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE



Records Properties

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Controlling the Appearance of Data Blocks

Setting Record Properties

Current Record Visual Attribute Group This group names a visual attribute that will be used to highlight the current record in the data block.

Query Array Size This size specifies the maximum number of records that Form Builder should fetch from the database at one time.

A lower value in this property value means faster response time; however, a larger value means fewer calls to the database for records, thereby resulting in reduced overall processing time.

Number of Records Buffered This is the minimum amount of buffer space retained for holding queried records in the data block. The minimum setting allowed is the value of the Number of Records Displayed property plus 3.

Form Builder buffers any additional records to a temporary disk file.

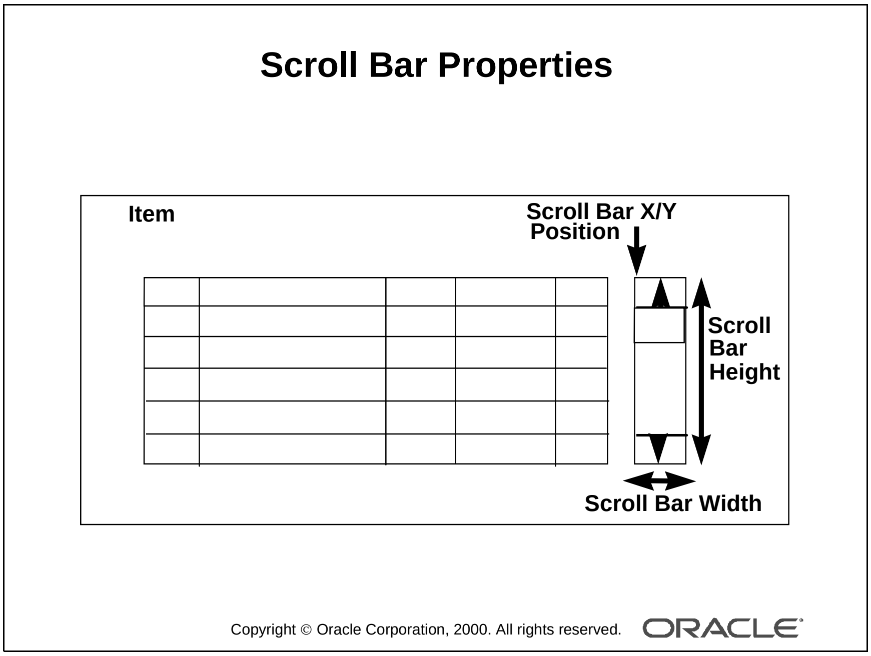
Number of Records Displayed This property specifies the maximum number of records the data block can display on the canvas at one time and how many records you can see at once. If you change this value, make sure there is enough room on the canvas layout for the number of records, or objects may overlap.

Query All Records This property specifies whether all the records matching the query criteria should be fetched when a query is executed. (This query is necessary to support the Calculated Field feature.)

Record Orientation This property determines the orientation of records in the data block—horizontal or vertical. When you set this property, Form Builder adjusts the display position of items in the data block accordingly.

Single Record This property specifies that the control block should always contain one record. Set this property to Yes for a control block that contains a summary calculated item.

Note: You cannot set this property to Yes for a data block.



Scroll Bar Properties

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

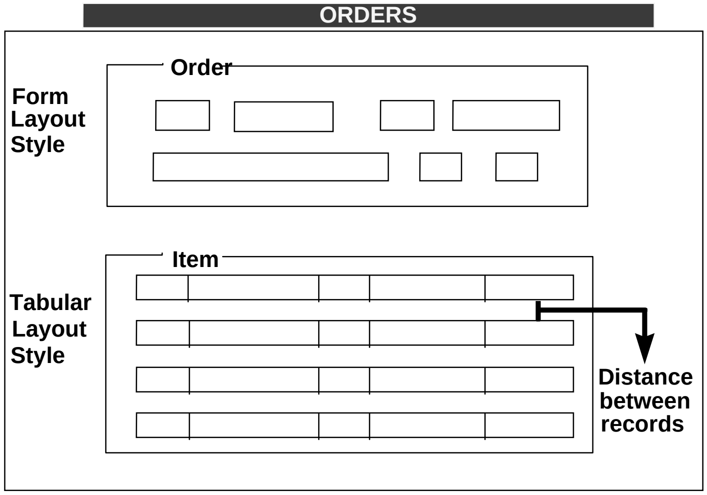
Improve processing speed by increasing the number of records buffered. Save memory by decreasing the number of records buffered. This can, however, result in slower disk I/O.

Demonstration This demonstration is for the previous page. Using the `ordwk04.fmb`, assign the visual attribute created earlier to the S_ITEM block. Show the run time effect of doing this. Also show how to increase or decrease the number of records displayed in a data block.

Setting Scroll Bar Properties

- Show Scroll Bar: This property specifies whether Form Builder should create a scroll bar for the data block. To delete an existing scroll bar, set this property to No.
- Scroll Bar Canvas: This property specifies the canvas on which the data block scroll bar will be displayed. The specified canvas must exist in the form.
- Scroll Bar Tab Page: This property specifies the canvas tab page on which the scroll bar will be displayed.
- Scroll Bar Orientation: This property specifies whether the scroll bar should be displayed horizontally or vertically.
- Scroll Bar X/Y Position: This property specifies the x and y coordinates (measured in the coordination system units of the form) where the scroll bar will display on the canvas. The default value for both coordinates is 0.
- Scroll Bar Width/Height: This property specifies the width and height of the scroll bar.
- Reverse Direction: This property specifies that the scroll bar scrolls in reverse. Setting this value to Yes causes Form Builder to fetch the next set of records when the user scrolls upward. If the user scrolls downward, Form Builder displays already fetched records.

Frame Properties



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Instructor Note

Stress the importance of the Update Layout property.

Controlling Frame Properties

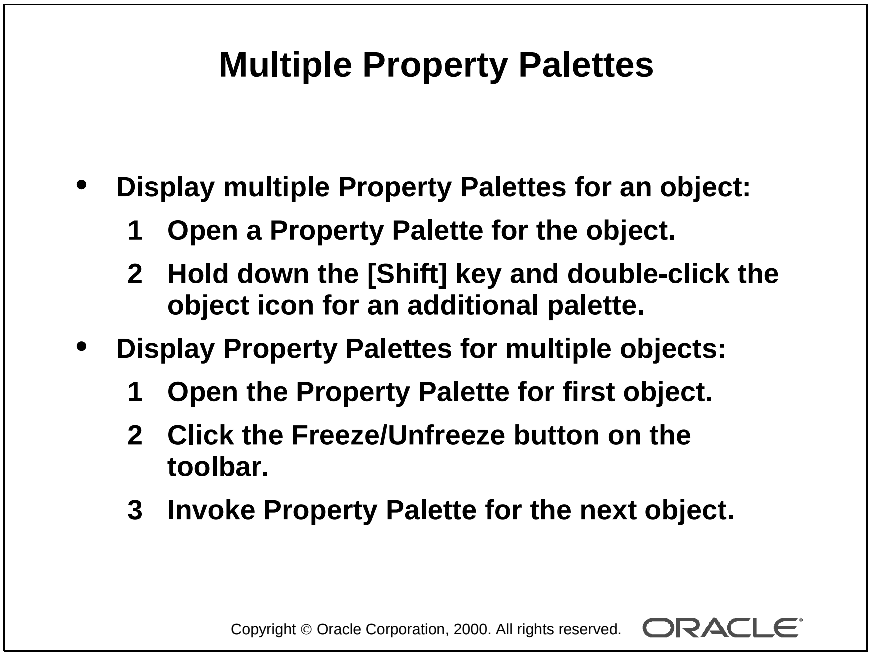
The selections that you make in the Layout Wizard when creating a data block are recorded as properties of the resulting layout frame object. You can change frame properties to modify the arrangements of items within a data block. The main frame properties are as follows:

Layout Data Block This property specifies the name of the data block with which the frame is associated. The items within this data block are arranged within the frame.

Note: A data block can be associated with only one frame. You cannot arrange a block item within multiple frames.

- Update Layout: Specifies when the frame layout is updated. Valid settings are:
  - Automatically: The layout is updated whenever you move or resize the frame, or modify any frame layout property.
  - Manually: The layout is updated whenever you use the Layout Wizard to modify the frame, or in the Layout Editor, when you click Update Layout or select the Arrange—>Update Layout menu option.
  - Locked: The layout is locked and cannot be updated.
- Layout Style: This property specifies the layout style for the items within the frame. Choose between Form and Tabular styles.
- Distance Between Records: This property specifies the physical distance (measured in the form’s coordination system units) with which to separate records displayed in the frame.
- X/Y Position: This property specifies the x and y coordinates (measured in the form’s coordination system units) of the frame’s position on the canvas.
- Width/Height: This property specifies the width and height of the frame (measured in the form’s coordination system units).

Note: You can arrange a frame as well as the objects within it manually in the Layout Editor.



Multiple Property Palettes

- **Display multiple Property Palettes for an object:**
  1. **Open a Property Palette for the object.**
  2. **Hold down the [Shift] key and double-click the object icon for an additional palette.**
- **Display Property Palettes for multiple objects:**
  1. **Open the Property Palette for first object.**
  2. **Click the Freeze/Unfreeze button on the toolbar.**
  3. **Invoke Property Palette for the next object.**

Copyright © Oracle Corporation, 2000. All rights reserved.

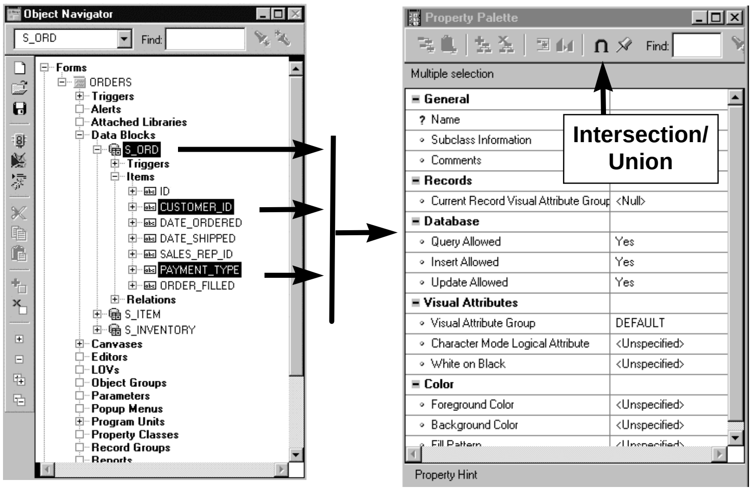
More About Object Properties

Displaying Multiple Property Palettes

- To display the properties of an object in multiple Property Palettes: While one Property Palette is already open, hold down the [Shift] key and double-click the object icon for the object in the Object Navigator.
- To display the Property Palettes for multiple objects at the same time:
  1. Open the Property Palette of the first object. Click Freeze/Unfreeze on the toolbar to “freeze” this palette.
  2. Invoke the Property Palette for another object. This Property Palette appears in a separate window.

If the second window is on top of the first one, drag it alongside the first window.

Setting Properties on Multiple Objects



Multiple selection

General	
Name	
Subclass Information	
Comments	
Records	
Current Record Visual Attribute Group	<Null>
Database	
Query Allowed	Yes
Insert Allowed	Yes
Update Allowed	Yes
Visual Attributes	
Visual Attribute Group	DEFAULT
Character Mode Logical Attribute	<Unspecified>
White on Black	<Unspecified>
Color	
Foreground Color	<Unspecified>
Background Color	<Unspecified>
Fill Pattern	<Unspecified>

Intersection/
Union

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Show how to select multiple objects in the Object Navigator, and how their combined properties can be represented in the Property Palette. Switch between Intersection and Union, pointing out the difference between the properties.

Setting Properties on Multiple Objects

You can view and set the properties of several objects simultaneously, whether they are the same or different object types. To do this, select the objects in the Object Navigator and display a combination of the properties in the Property Palette. The combination or set may be:

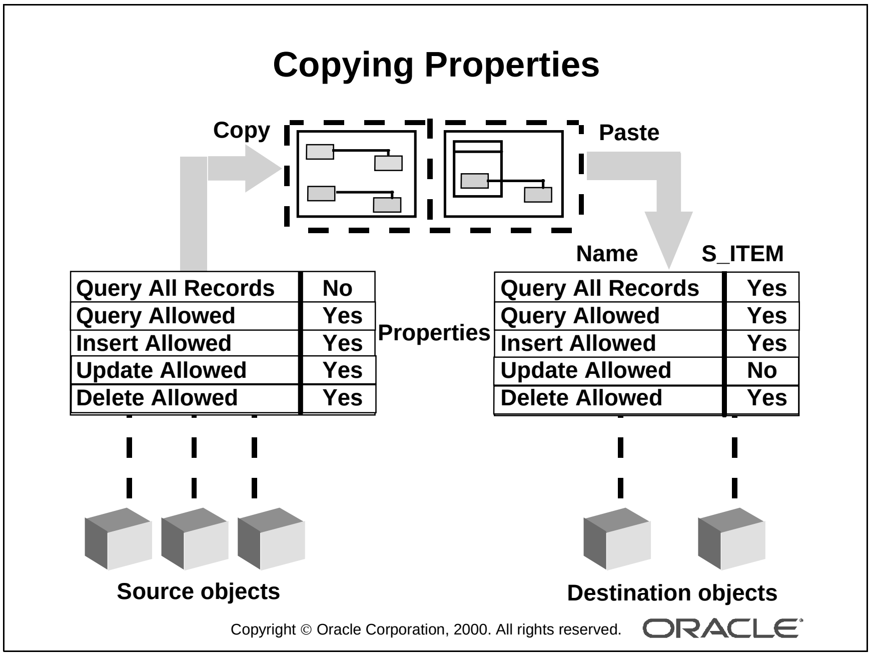
- Intersection (∩): A subset in which you display only the common properties of the selected objects (This is the default set operator.)
- Union (∪): A superset in which you display both the common properties and the unique properties of the selected objects

Where there are differing values for a property across the selected objects, you will see ***** in the property value. This changes to a definitive value once you enter a new value in the Property Palette. This new value then applies to each of the selected objects to which the property is relevant.

How to Set Properties on Multiple Objects

1. Open the Property Palette for one of the objects.
2. Hold down the [Ctrl] key and click each object in the Object Navigator or the editors whose properties are to be viewed or changed in combination. The selected objects are highlighted.
3. Set the Intersection/Union button from the toolbar in the Property Palette to the desired operation. This button toggles between the two options.
4. Change the displayed properties, as required. Your changes are applied to all selected objects with these properties.

Note: With a union, some properties you see might not be relevant to all of the selected objects. Changes to a property are applied only to objects that have the property.



Copying Properties

Query All Records	No
Query Allowed	Yes
Insert Allowed	Yes
Update Allowed	Yes
Delete Allowed	Yes

Query All Records	Yes
Query Allowed	Yes
Insert Allowed	Yes
Update Allowed	No
Delete Allowed	Yes

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Show how to copy the properties of one object to another. Do not save the effects of this demonstration.

Note: Property inheritance and variance are covered in a later lesson.

How to Copy Properties to Other Objects

You can write the properties and values from the Property Palette to a buffer, so that they can be applied (pasted) to other objects in your design session.

1. In the Property Palette, display and set the properties to be copied. This may be from one object or a combination of them.
   - To copy all the property settings from the Property Palette, select Edit—>Select All.
   - To copy selected property settings only, hold down the [Ctrl] key and click each property individually.
2. Click Copy Properties on the toolbar of the Property Palette.
3. From the Object Navigator select the object into which the properties are to be copied.
4. In the Property Palette, click Paste Properties. The selected object receives values from all copied properties that are relevant to their object types.

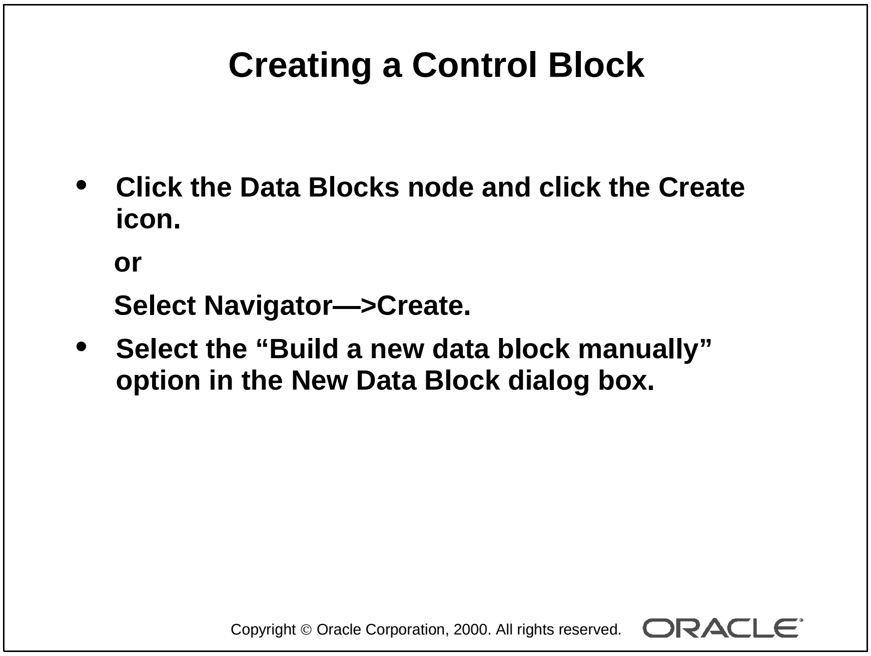
Note: It is possible to copy the property settings of an object to objects of different types. In this case, properties that do not apply to the target object are ignored.

Property Classes

When you display a list of properties (from either one object or a combination of objects) in the Property Palette, the list of property names and associated values can be saved for future application to other objects. This is known as a property class, which is a Form Builder object in its own right.

Objects can inherit some of their properties from a linked property class, so their properties will automatically change if the associated properties are changed in the property class.

Property classes are discussed in more detail in a later lesson.



Creating a Control Block

- **Click the Data Blocks node and click the Create icon.**

  or

  Select Navigator—>Create.
- **Select the “Build a new data block manually” option in the New Data Block dialog box.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Create a control block in the form module, using the `ordwk04.fmb` file.

Creating Control Blocks

A control block is a block that is not associated with any database, and its items do not relate to any columns within any database table.

This means that Form Builder does not perform an automatic query when the operator issues an Enter Query or Execute Query command, nor does it issue an automatic Insert, Update, or Delete for the block when the operator saves changes to the database.

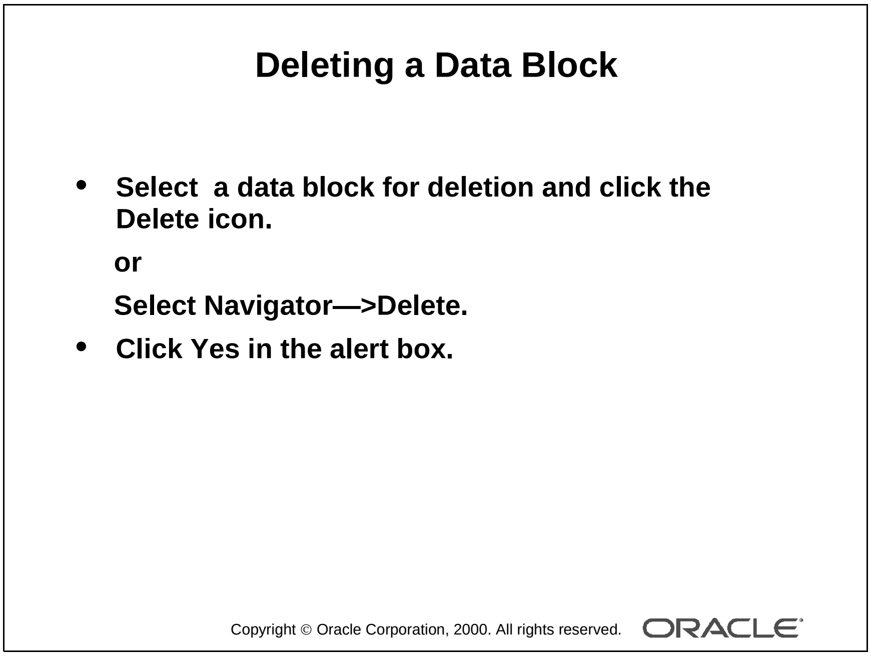
How to Create a Control Block

1. Click the Data Blocks node in the Object Navigator.
2. Click the Create icon on the toolbar.

   or

   Select Navigator—>Create from the menu.
3. In the New Data Block dialog box, select the “Build a new data block manually” option.
4. Open the Property Palette of the new data block and change its name.

Note: Because there are no database columns on which to base control block items, a control block has no items until you manually add them later.



Deleting a Data Block

- Select a data block for deletion and click the Delete icon.

  or

  Select Navigator—>Delete.
- Click Yes in the alert box.

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration   Delete the S_ITEM data block in the `ordwk04.fmb` file. Show the effect of doing this in the Layout Editor. Do not save the effects of this demonstration.

Deleting Data Blocks

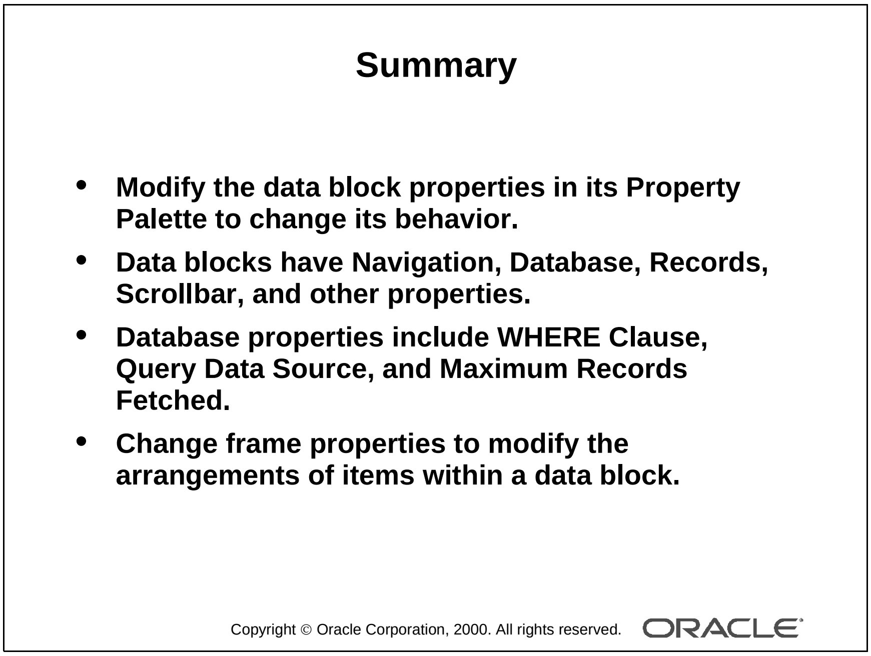
To delete a data block:

1. Select the data block to be deleted in the Object Navigator.
2. Click the Delete icon on the toolbar.

   or

   Select Navigator—>Delete from the menu.
3. An alert is displayed for delete confirmation. Click Yes to delete the data block.

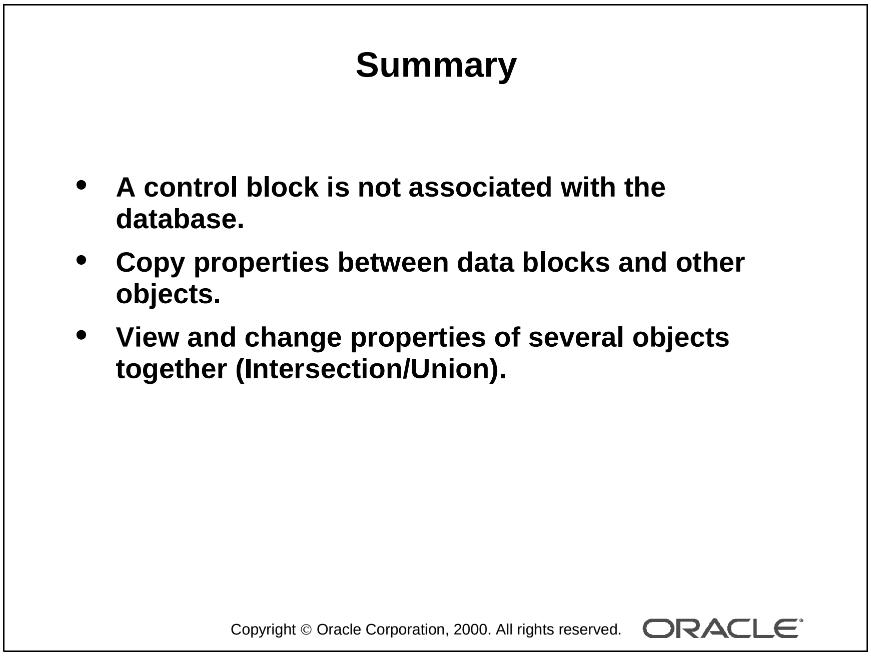
Note: Deleting a data block also deletes its subordinate objects (items and triggers). If the data block was a master or detail block in a relation, the relation is also deleted. However, the frame border and its title will remain. Delete the frame manually in the Layout Editor.



Summary

- **Modify the data block properties in its Property Palette to change its behavior.**
- **Data blocks have Navigation, Database, Records, Scrollbar, and other properties.**
- **Database properties include WHERE Clause, Query Data Source, and Maximum Records Fetched.**
- **Change frame properties to modify the arrangements of items within a data block.**

Copyright © Oracle Corporation, 2000. All rights reserved.



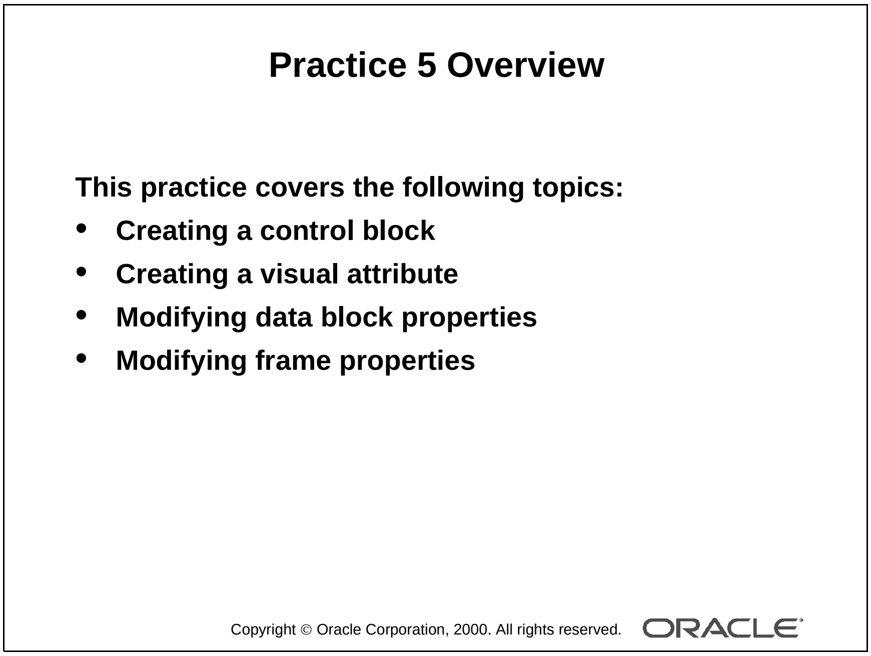
Summary

- **A control block is not associated with the database.**
- **Copy properties between data blocks and other objects.**
- **View and change properties of several objects together (Intersection/Union).**

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

- Modify the data block properties in its Property Palette to change its behavior at run time.
- Data blocks have Navigation, Database, Records, Scrollbar, and other properties.
- Database properties include WHERE Clause, Query Data Source Type, and Maximum Records Fetched.
- You can change frame properties to modify the arrangements of items within a data block.
- You can copy properties between data blocks and other objects.
- You can view and change the properties of several objects together. You can use Intersection or Union settings to connect their properties in the Property Palette.



Practice 5 Overview

This practice covers the following topics:

- **Creating a control block**
- **Creating a visual attribute**
- **Modifying data block properties**
- **Modifying frame properties**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 5 in Appendix A, “Practice Solutions.”

Practice 5 Overview

In this practice session, you will create control blocks and modify several data block properties as well as frame properties.

- Create a control block in the CUSTOMERS form.
- Using the Property Palette, change properties in the S_CUSTOMER data block to change its run-time appearance and behavior. Change the related frame properties to change the run-time appearance and to keep any layout changes you make manually in the Layout Editor. Save and run the form after the changes are applied.
- Create a control block in the ORDERS form.
- Create a visual attribute in the ORDERS form and use it to highlight the current record in the S_ITEM and S_INVENTORY data blocks at run time. Use the multiple selection feature on both data blocks when setting the appropriate property in the Property Palette.
- Change properties in the S_ITEM and S_INVENTORY data blocks to change their run-time appearance and behavior. Change the frame properties of all the data blocks in the ORDERS form to change their run-time appearance and to keep any layout changes you make manually in the Layout Editor. Save and run the form after the changes are applied.

Practice 5

CUSTG*XX* Form

1 Create a control block in the CUSTG*XX* form.

Create a new block manually, and rename this block CONTROL.

Set the Database Data Block, Query Allowed, Insert Allowed, Update Allowed, and Delete Allowed Database properties to No. Set the Query Data Source Type property to None. Leave other properties as default.

Move the CONTROL block after the S_CUSTOMER block.

2 Ensure that the records retrieved in the S_CUSTOMER block are sorted by the customer’s ID.

3 Set the frame properties for the S_CUSTOMER block as follows:

Remove the frame title, and set the Update Layout property to Manually.

4 Save and run the CUSTG*XX* form.

Test the effects of the properties that you have set.

Note: The Compilation Errors window displays a warning that advises you that the CONTROL block has no items. This is expected (until you add some items to the CONTROL block in a later lesson).

ORDG*XX* Form

5 Create a control block in the ORDG*XX* form.

Create a new block manually, and rename this block CONTROL.

Set the Database Data Block, Query Allowed, Insert Allowed, Update Allowed, and Delete Allowed database properties to No. Set the Query Data Source Type property to None. Leave other properties as default.

Position the CONTROL block after the S_INVENTORY block in the Object Navigator.

Hint: You can copy the CONTROL block from the CUSTG*XX* form.

ORDG*XX* Form (continued)

6 Ensure that the current record is displayed differently from the others in both the S_ITEM and S_INVENTORY blocks.

Create a visual attribute called Current_Record.

Using the Color Picker, set the Foreground Color to White and the Background Color to Dark Cyan. (If these colors are not available on your window manager, use other colors instead.) Using the Pattern Picker, choose any fill pattern. Using the Font Picker, set the font to MS Serif italic 10 point. (If that font is not available on your window manager, use any available font.)

Use the multiple selection feature on both data blocks to set the relevant block property to use this visual attribute.

7 For the S_ITEM block, change the number of records displayed to 4 and resize the scroll bar accordingly.

8 Ensure that the records retrieved in the S_ITEM block are sorted by the ITEM_ID.

9 Set the property that causes automatic navigation to Next Record, when the user uses [Next Item] to exit the last item of a record in the S_ITEM block.

10 Set the frame properties for all blocks as follows:

Remove the frame title and set the Update Layout property to Manually.

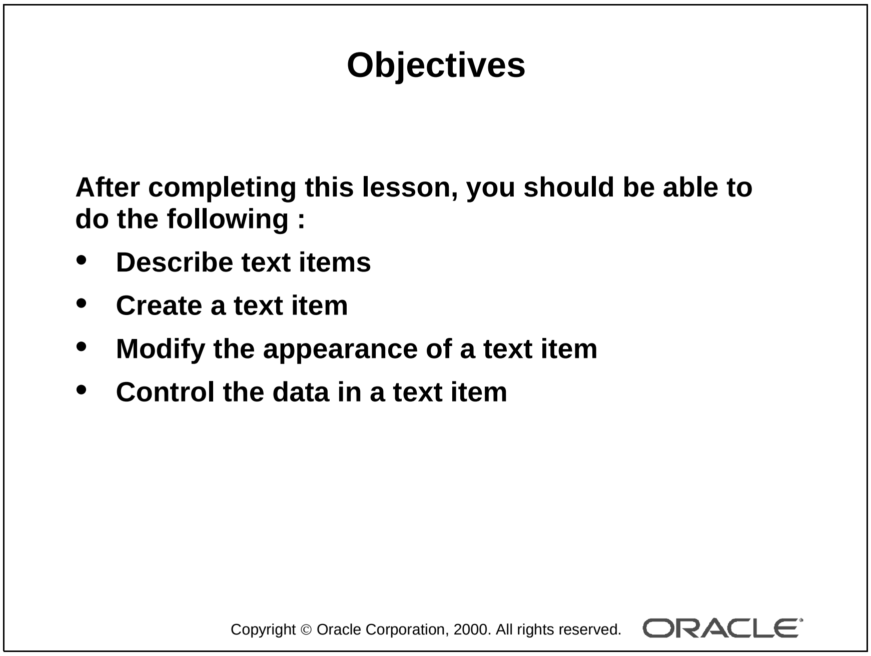
11 Save and compile the ORDG*XX* form.

Deploy your form on the Web.

Test the effects of the properties that you have set.

6

Working with Text Items

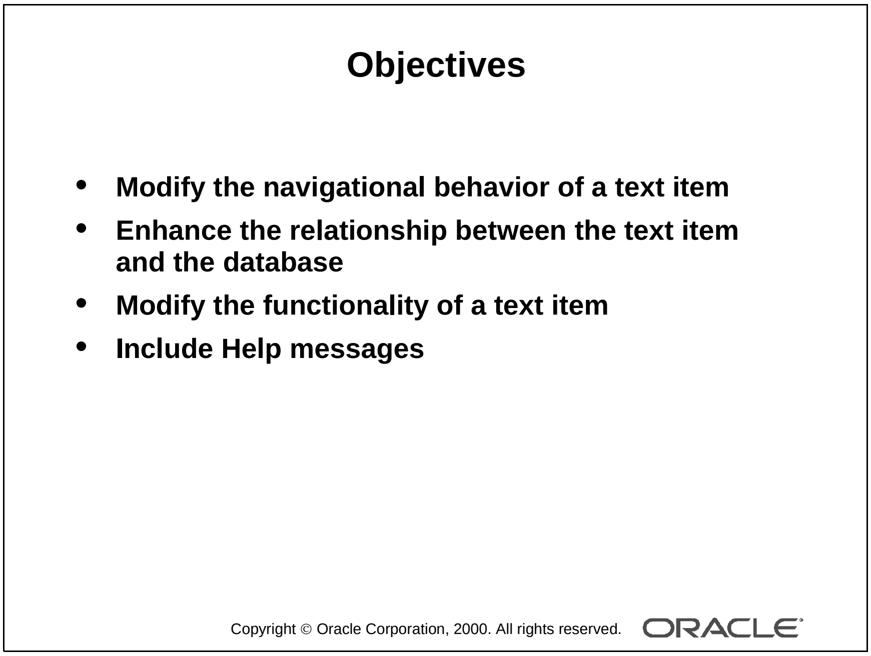


Objectives

After completing this lesson, you should be able to do the following :

- **Describe text items**
- **Create a text item**
- **Modify the appearance of a text item**
- **Control the data in a text item**

Copyright © Oracle Corporation, 2000. All rights reserved.



Objectives

- **Modify the navigational behavior of a text item**
- **Enhance the relationship between the text item and the database**
- **Modify the functionality of a text item**
- **Include Help messages**

Copyright © Oracle Corporation, 2000. All rights reserved.

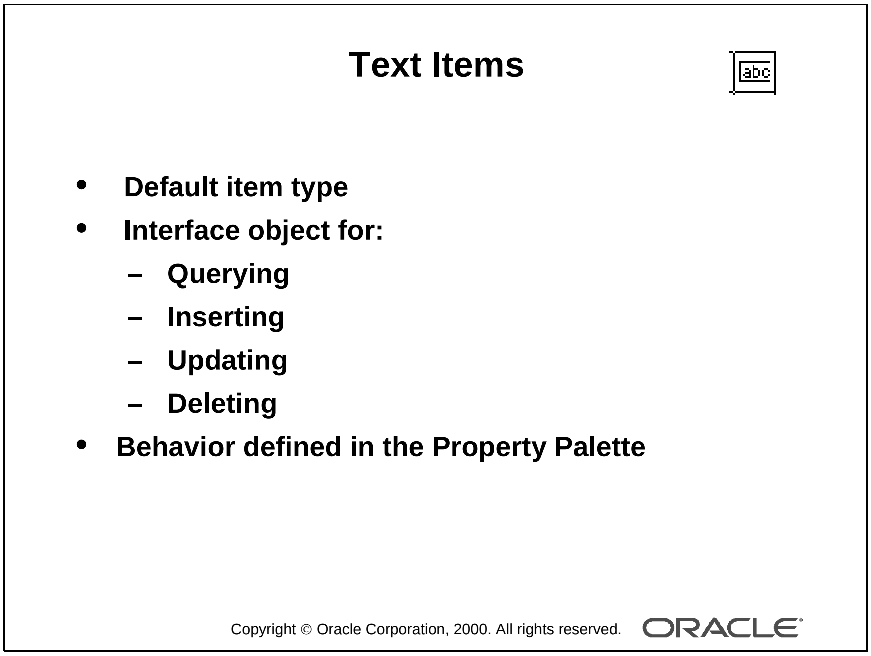
Schedule

Topic	**Timing**
Lecture	60 minutes
Practice	40 minutes
Total	100 minutes

Introduction

Overview

The default item type in an Oracle Forms Developer application is the text item or field. You have seen how creating a new data block based on a table creates text items for each selected column from that table. This lesson shows you how to customize text items to change their appearance and behavior.



Text Items

- **Default item type**
- **Interface object for:**
  - **Querying**
  - **Inserting**
  - **Updating**
  - **Deleting**
- **Behavior defined in the Property Palette**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

- Although the technical term *item* is used in design terminology, end users see these objects as *fields.*
- Ensure that the class does not get confused between *item* as a Form Builder term for the objects in a block, and *item* as the application term for line item records in the Order Entry system.

What Is a Text Item?

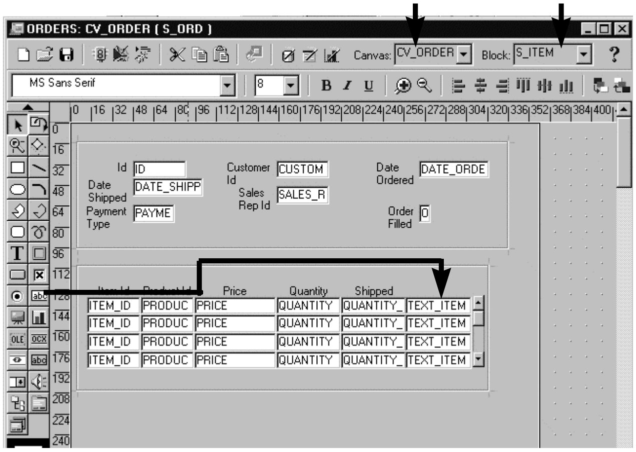
A *text item* is an interface object through which you can query, insert, update, and delete data. A text item usually corresponds to a column in the database table. When an item is first created, its default type is text.

The item type determines the properties available in the Property Palette. In this lesson you look at the properties of a text item. Remaining item types are covered in subsequent lessons.

Use the Property Palette to define, alter, or examine the characteristics of items.

Creating a Text Item

Canvas selection Block selection


ORDERS: CV_ORDER (S_ORD)
Canvas: CV_ORDER
Block: S_ITEM
MS Sans Serif
8
Id ID
Date Shipped DATE_SHIPP
Payment Type PAYME
Customer Id CUSTOM
Sales Rep Id SALES_R
Date Ordered DATE_ORDE
Order Filled 0
Item Id
Product Id
Price
Quantity
Shipped
ITEM_ID PRODUC PRICE QUANTITY QUANTITY_ TEXT_ITEM
ITEM_ID PRODUC PRICE QUANTITY QUANTITY_ TEXT_ITEM
ITEM_ID PRODUC PRICE QUANTITY QUANTITY_ TEXT_ITEM
ITEM_ID PRODUC PRICE QUANTITY QUANTITY_ TEXT_ITEM

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration

- In the first demonstration, show how to create a text item in the Layout Editor.
- In the second demonstration, create a text item in the Object Navigator and do the following:
  - Go to the Layout Editor. The new text item is not seen. Point out to the students that this is a Null canvas item, because it is not assigned to any canvas yet.
  - Open the text item property sheet, and assign the Canvas property to the desired canvas.
  - Show the text item in the Layout Editor.
- Point out the importance of creating a new text item in the correct data block and canvas. Also point out the Canvas and Block pop-up lists in the Layout Editor.

Creating a Text Item

You can create a text item by doing one of the following:

- Converting an existing item into a text item
- Using the Text Item tool in the Layout Editor
- Using the Create icon in the Object Navigator
- Using the wizards

How to Create a Text Item in the Layout Editor

1. Invoke the Layout Editor.

   It is important to point to the correct data block where you want to create the text item. In the Layout Editor, select the data block from the Block pop-up list.
2. Click the Text Item tool.
3. Click the canvas.

   The text item appears.
4. Double-click the text item.

   The text item Property Palette appears.
5. Set the item properties as required.

How to Create a Text Item in the Object Navigator

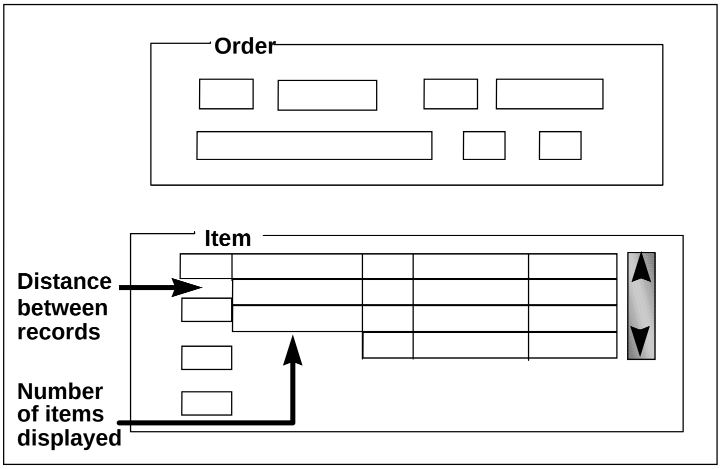
1. Locate the block in which you want to create the item.
2. Select the Items node.
3. Click the Create icon.

   A new item entry is displayed in the Object Navigator.
4. Double-click the icon to the left of the new item entry.

   The Property Palette appears.
5. Set the Type property to Text Item.
6. Set all other item properties as required.

Note: To display an item at run time, you must assign the item to a canvas. Do this in the Property Palette of the text item by setting the Canvas property to the desired canvas.

Modifying the Appearance of a Text Item



Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Using the `ordwk05.fmb` file, show how to alter the width and height of a text item. Show how to change the number of items displayed at run time.

Modifying the Appearance of a Text Item

The properties of an item are divided into several groups.

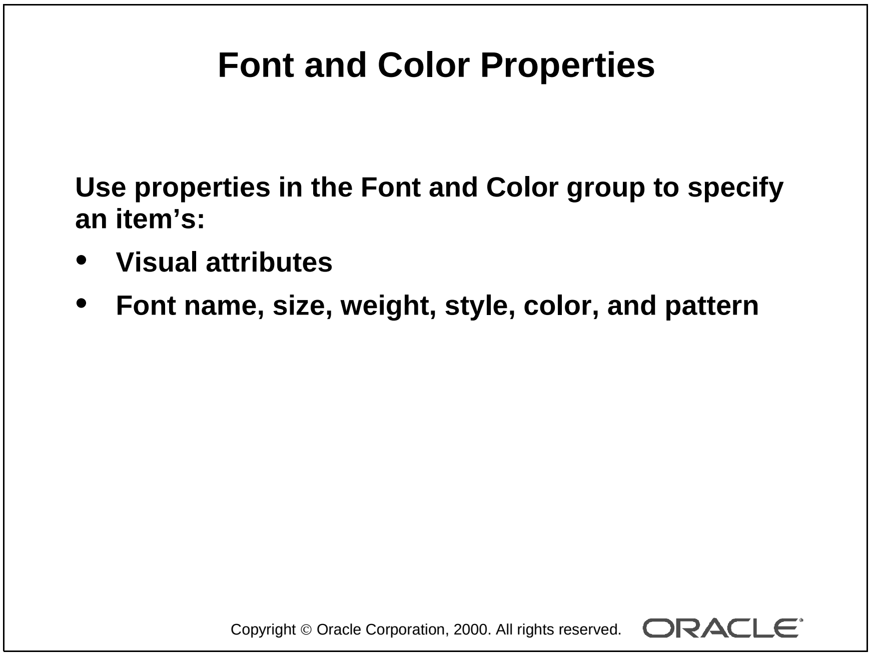
You can affect the way the text item is displayed by altering its General, Physical, Records, and Font and Color group properties.

General Property	**Function**
Item Type	Selects the type of item you want to create

Physical Property	**Function**
Visible	Determines whether the item is displayed
Canvas	Determines on which canvas the item is displayed
Tab Page	Determines the tab page on which the item is located
X Position	Sets the X coordinate of the item relative to the canvas (bitmapped) or the screen (character)
Y Position	Sets the Y coordinate of the item relative to the canvas (bitmapped) or the screen (character)
Width	Sets the width of the text item in the current form coordinate units
Height	Sets the height of the text item in the current form coordinate units
Bevel	Sets the sculpted appearance of the item border
Rendered	Conserves system resources (When a rendered item no longer has focus, the resources used to display it are released.)
Show Vertical Scrollbar	Determines whether a vertical scroll bar appears for the multiline text item

Records Property	**Function**
Current Record Visual Attribute Group	Specifies the name of the visual attribute to use when the item is part of the current record
Distance Between Records	Specifies the amount of space between instances of the item in a multirecord data block
Number of Items Displayed	Specifies the number of item instances displayed for the item when the item is in a multirecord block

Note: When the Canvas property of an item is left unspecified, it is said to be a *Null canvas* item. It will not display at run time or in the Layout Editor.



Font and Color Properties

Use properties in the Font and Color group to specify an item’s:

- **Visual attributes**
- **Font name, size, weight, style, color, and pattern**

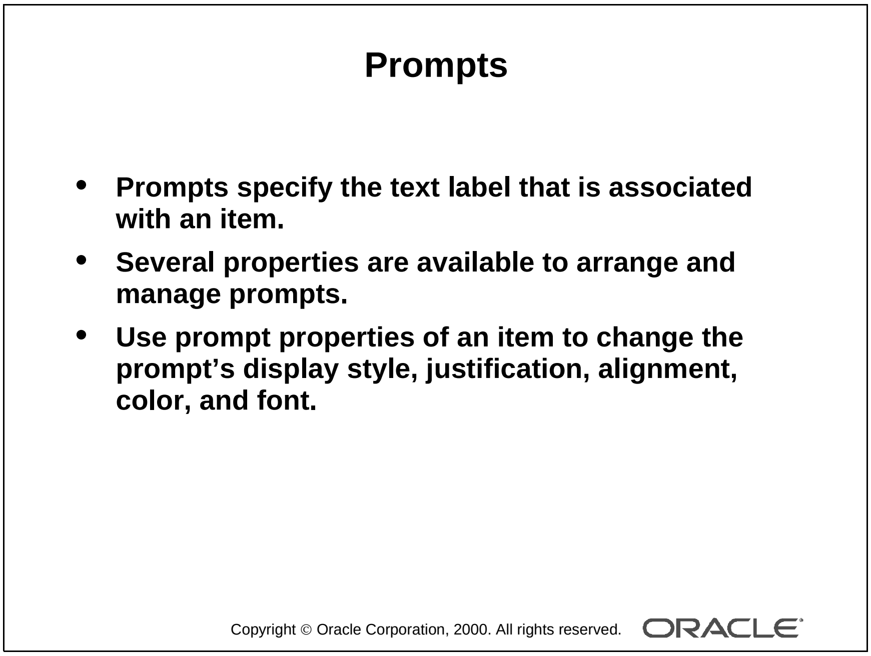
Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

When the form module does not contain any named visual attribute objects, the pop-up list for the Visual Attribute Group property shows only Default.

An item that has the Visual Attribute Group property set to default, or that has individual attribute settings left unspecified, inherits those settings from the canvas to which it is assigned.

Font and Color Property	**Function**
Visual Attribute Group	Specifies how the item’s individual visual attributes are derived (Select Default or Named.)
Font Name	Specifies the font family (The list of fonts available is system dependent.)
Font Size	Specifies the font size in points
Font Weight	Specifies the weight of the font
Font Style	Specifies the font style
Font Spacing	Specifies the width of the font (This is the amount of space between characters.)
Foreground Color	Specifies the foreground color for the item
Background Color	Specifies the background color for the item
Fill Pattern	Specifies the pattern for the item fill region
Character Mode Logical Attribute	Specifies the name of the Oracle Terminal resource file attribute that sets the attributes of the device when running in character mode
White on Black	Determines whether the item appears as white text on a black background when displayed on a monochrome device



Prompts

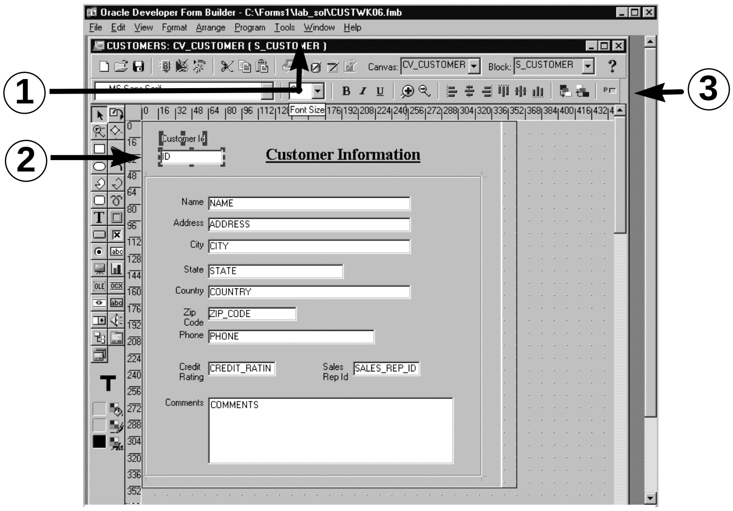
- **Prompts specify the text label that is associated with an item.**
- **Several properties are available to arrange and manage prompts.**
- **Use prompt properties of an item to change the prompt’s display style, justification, alignment, color, and font.**

Copyright © Oracle Corporation, 2000. All rights reserved.

Prompt Property	**Function**
Prompt	Specifies the text label for the item
Prompt Display Style	Specifies the display style of the prompt (Choose among First Record, Hidden, and All Records.)
Prompt Justification	Specifies how the prompt is justified
Prompt Attachment Edge	Specifies the item edge to which the prompt is attached
Prompt Alignment	Specifies how the prompt is aligned along the item edge (Choose among: Start, End, and Center.)
Prompt Attachment Offset	Specifies the distance between the item and its prompt
Prompt Alignment Offset	Specifies the alignment offset of the prompt
Prompt Reading Order	Specifies the prompt order (Choose among: Default, Left To Right, and Right To Left.)

Prompt Font and Color Properties	**Function**
Prompt Visual Attribute Group	Specifies the named visual attribute that should be applied to the item prompt at run time
Prompt Font Name	Specifies the font family or typeface of the item prompt
Prompt Font Size	Specifies the size of the prompt font, in points
Prompt Font Weight	Specifies the weight of the prompt font
Prompt Font Style	Specifies the style of the prompt font
Prompt Font Spacing	Specifies the amount of space between characters
Prompt Foreground Color	Specifies the prompt text color

Associating Text with an Item Prompt



Copyright © Oracle Corporation, 2000. All rights reserved.

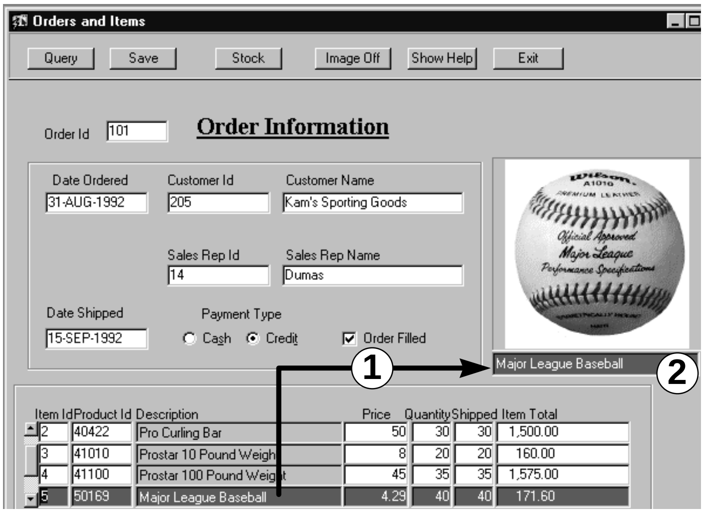
Associating Text with an Item Prompt

The enhanced Form Builder Layout Editor has a tool called Associate Prompt which enables you to create a prompt for an item using any boilerplate text in the editor. In order to create a prompt-item association using the Associate Prompt tool, do the following:

1. Open the Layout Editor window.
2. Select the item and boilerplate text you want as the item’s prompt in the editor.
3. Click the Associate Prompt tool.

Note: Users are able to create text items *with* a border but *without* a bevel by setting the Bevel property to Plain. In releases prior to Release 6.0, if bevel for a text item was removed, the border was also removed.

Synchronize with Item Property



Order Id 101

Order Information

Date Ordered	Customer Id	Customer Name
31-AUG-1992	205	Kam's Sporting Goods

Sales Rep Id	Sales Rep Name
14	Dumas

Date Shipped 15-SEP-1992

Payment Type ○ Cash ● Credit

☑ Order Filled

Major League Baseball

Item Id	Product Id	Description	Price	Quantity	Shipped	Item Total
2	40422	Pro Curling Bar	50	30	30	1,500.00
3	41010	Prostar 10 Pound Weigh	8	20	20	160.00
4	41100	Prostar 100 Pound Weight	45	35	35	1,575.00
5	50169	Major League Baseball	4.29	40	40	171.60

Copyright © Oracle Corporation, 2000. All rights reserved.

1	Synchronize with Item property set to Description item
2	Number of Items Displayed property set to 1

Instructor Note

- The number of characters a text item can display is determined by a combination of factors, including the width of the text item, the font attributes, and the run-time platform.
- Form Builder assigns a default value whenever the item is part of a record whose status is marked as NEW (when a new record is created).
- If a text item is based on a column with a NOT NULL constraint, Form Builder automatically sets the Required property of the item to Yes.
- You can use the Synchronize with Item property to create more than one item in a data block that displays the same database column values.
- When data blocks are related through a compound join, the Copy Value from Item property is set on two or more foreign-key items in the detail data block.

Controlling the Data of a Text Item

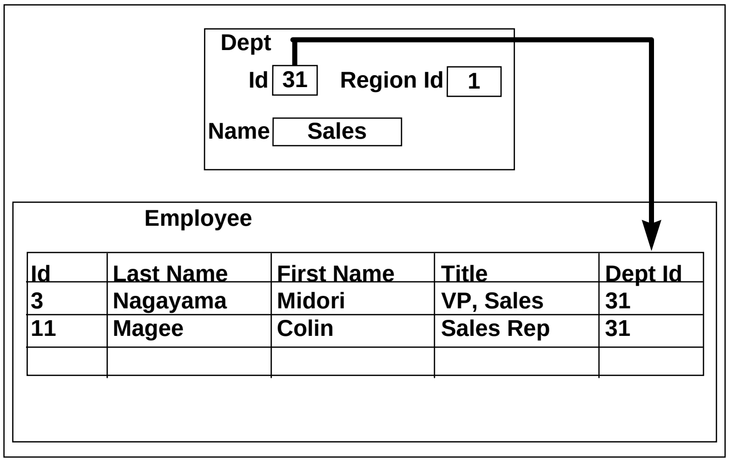
Use the data group properties of a text item to control the way the data is displayed and entered.

Data Property	**Function**
Data Type	Determines what kinds of values Form Builder allows as input into the item
Maximum Length	Sets the maximum number of characters allowed in the item (This usually corresponds to the column width for base table items.)
Fixed Length	Determines whether the user is required to fill the item completely before being able to move into another item
Initial Value	Specifies the default value that Form Builder gives to the item for each new record
Required	Determines whether the item is mandatory; that is, whether it is based on a NOT NULL column constraint in the underlying table
Format Mask	Specifies a format for the text item
Lowest Allowed Value	Sets the minimum value allowed in this item
Highest Allowed Value	Sets the maximum value allowed in this item
Copy Value from Item	Specifies the block and item to use as the source for copying a value into this item when focus is set on one of its records (used in interblock relations)
Synchronize with Item	Specifies the name of the item from which the current item derives its value (This property means that you can have two items in a block that simultaneously represent the same base table column, thus mirroring each other.)

Note

- If the Maximum Length exceeds the display width of the item, Form Builder automatically enables the end user to scroll the contents of the item horizontally, so that the cursor remains visible.
- When Fixed Length is set to Yes, a text item is valid only if it contains the number of characters specified in the Maximum Length property.
- Form Builder accepts NULL values in a fixed length text item, provided the text item does not have its Required property set to Yes.
- Use only CHAR, NUMBER, DATE, and LONG data types. All other data types are available for compatibility with earlier versions of Oracle Forms Developer.

Copy Value from Item Property



Dept

Id 31 Region Id 1

Name Sales

Employee

Id	Last Name	First Name	Title	Dept Id
3	Nagayama	Midori	VP, Sales	31
11	Magee	Colin	Sales Rep	31

Copyright © Oracle Corporation, 2000. All rights reserved.

Format Masks

- **Standard SQL formats**
  - **Dates FXDD-MON-YY**
  - **Numbers L099G99D99**
- **Nonstandard formats**

  Use double quotes for embedded characters

  "("099")"099"-"0999

Note: Allow for format mask's embedded characters when defining Width property.

Copyright © Oracle Corporation, 2000. All rights reserved.

The Copy Value from Item Property

Use the Copy Value from Item property to reference the source of the value that Forms uses to populate the current item.

- Enter the block and item name from which Form Builder reads the value as follows:

  `<data_block_name>.<item_name>`.
- Use this property to link data blocks. In a master-detail form module, Form Builder automatically sets this property for the foreign-key item in the detail data block to the primary-key item in the master data block.
- In the Summit application, S_ITEM.ORD_ID has this property.

Note: The text item should disable input; otherwise, the user could violate the foreign-key relationship. To prevent operators from deactivating the foreign-key relationship, set the Enabled property to No for the foreign-key item.

Formatting a Text Item

Use the Format Mask property to specify the format in which the user sees the item value.

- Use standard SQL formatting syntax for dates and numbers; for example, DD/MM/YY and $99,999.99.
- Enclose non-SQL standard embedded characters in double quotes; for example, hyphen (-) and comma (,).

Note: It is recommended that you avoid creating individual masks if the general purpose masks (see Lesson 1) will suffice.

FX Format Mask

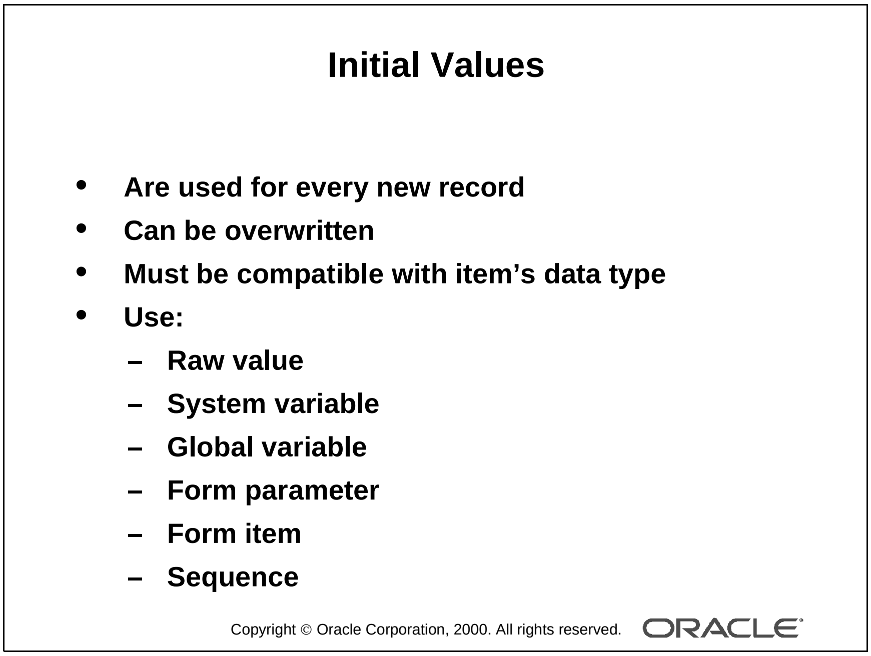
The FX format mask in a date value ensures that the date is entered exactly as defined in the mask. Element D is for decimal, and G is a group separator.

Example

With a date format of DD/MM/YY, valid entries are:

- 10/12/00
- 10 12 00
- 10-DEC-00
- 101200

Note: You can enter any character to represent the (/) in the value. Allow for the embedded characters of the format mask when defining the Width property. The embedded characters are used only for purposes of display and are not stored in the database.



Initial Values

- Are used for every new record
- Can be overwritten
- Must be compatible with item’s data type
- Use:
  - Raw value
  - System variable
  - Global variable
  - Form parameter
  - Form item
  - Sequence

Copyright © Oracle Corporation, 2000. All rights reserved.

Initial Values

The initial value must be compatible with the item data type. If the Lowest/Highest Allowed values are specified, the initial value cannot be outside the range.

Creating an Initial Value

Use any one of the following values to issue an initial item value whenever a new record is created:

- Raw value

  Example: 340, RICHMOND
- System variable
  - Variables giving current *operating system* date/time:

Variable	**Format**
$$DATE$$	DD-MON-YY
$$DATETIME$$	DD-MON-YYYY hh:mi[:ss]
$$TIME$$	hh:mi[:ss]

  - Variables giving current *database* date/time:

Variable	**Format**
$$DBDATE$$	DD-MON-YY
$$DBDATETIME$$	DD-MON-YYYY hh:mi[:ss]
$$DBTIME$$	hh:mi[:ss]

- Global variable

  Example: :GLOBAL.CUSTOMER_ID
- Form parameter

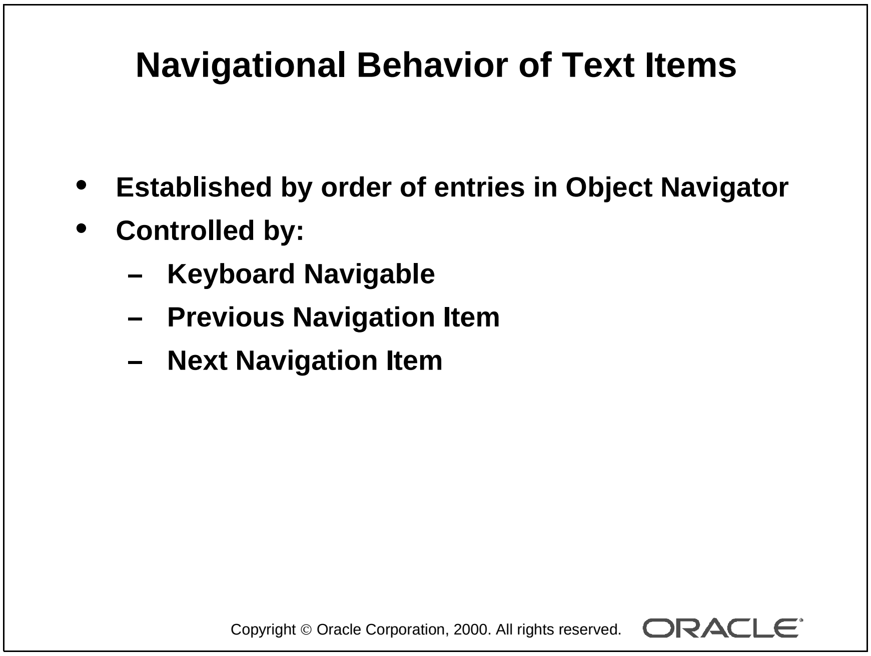
  Example: :PARAMETER.SALES_REP_ID
- Form item

  Example: :ORDER.ID
- Sequence

  Example: :SEQUENCE.S_ORD_ID.NEXTVAL

Automatic Sequence Number Generation

The initial value can reference a sequence in the database. Form Builder automatically writes generated sequence numbers into the text item.



Navigational Behavior of Text Items

- **Established by order of entries in Object Navigator**
- **Controlled by:**
  - **Keyboard Navigable**
  - **Previous Navigation Item**
  - **Next Navigation Item**

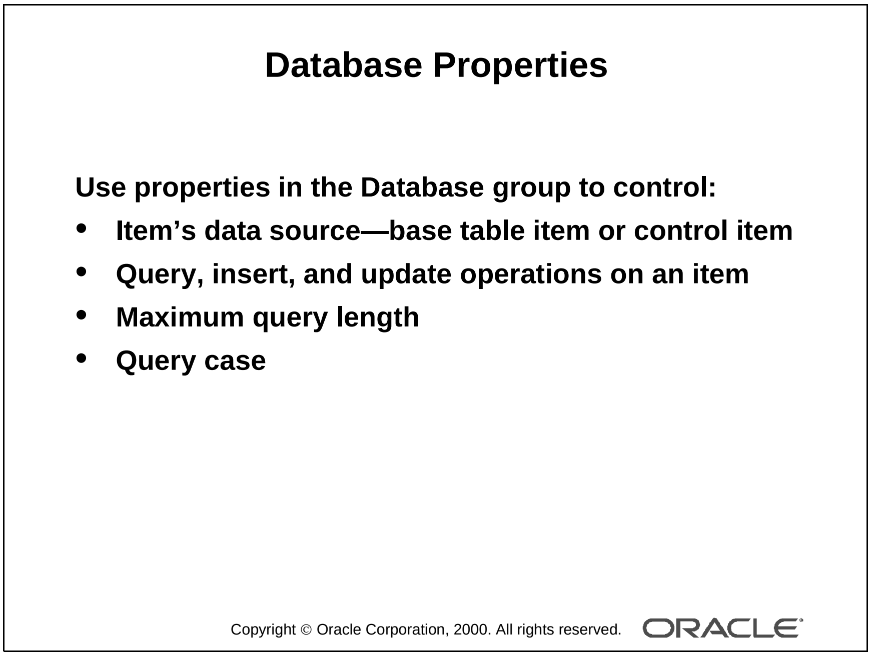
Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Altering the Navigational Behavior of a Text Item

You can see the default navigational sequence of items in the Object Navigator, as the item entries are displayed in the navigational order. However, you can also use the Navigation group properties to control the navigational behavior of a text item.

Navigation Property	**Function**
Keyboard Navigable	Determines whether you can navigate to an item during default navigation with the function keys or menu items and place input focus on it (When this property is set to No, Form Builder skips over the item and enters the next navigable item in the default navigation sequence.)
Previous Navigation Item	Determines the previous item to be visited when you navigate out of the current item
Next Navigation Item	Determines the next item to be visited when you navigate out of the current item

Note: The next or previous navigation item must be in the same data block as the current item.



Database Properties

Use properties in the Database group to control:

- **Item’s data source—base table item or control item**
- **Query, insert, and update operations on an item**
- **Maximum query length**
- **Query case**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

- Setting Case Insensitive Query to Yes may take queries longer to execute.
- Update Allowed must be set to No before setting Update Only if NULL to Yes.
- When might Update Only if NULL be used?

  For a value that can be set only once. For example, it might be used for a ship date. Once an order is shipped the ship date cannot be modified.

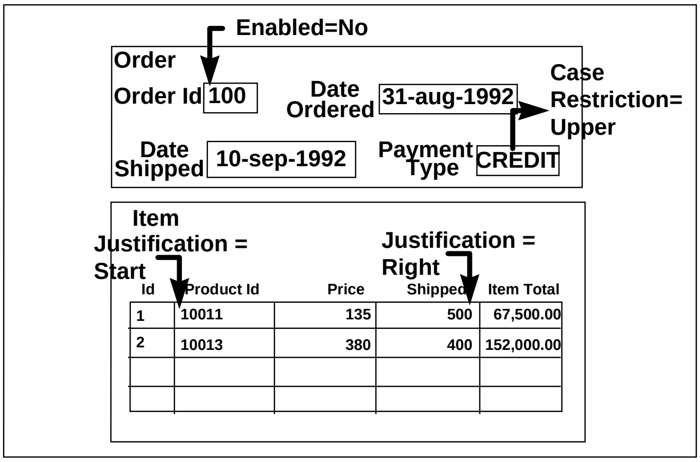
Enhancing the Relationship Between Text Item and Database

You can alter or enhance the way in which a text item interacts with its corresponding database column by setting the Database group properties.

Database Property	**Function**
Database Item	Determines whether the item value is stored in the block base table as opposed to being a control item
Column Name	Establishes that the item corresponds to a column in the table associated with the data block
Primary Key	Indicates that the item is a base table item and that it corresponds to a primary key column in the base table (Its value must be unique.)
Query Only	Specifies that the item can be queried but not included in an INSERT or an UPDATE statement
Query Allowed	Determines whether the item can accept query criteria
Query Length	Sets the maximum length of a query expression for the item (Enter Query mode) (This may not be set to a value lower than that of the Maximum Length, unless set to zero.)
Case Insensitive Query	Determines whether the difference between upper, mixed, and lowercase values is recognized during query processing
Insert Allowed	Determines whether the item allows values to be inserted (This property applies only to new records.)
Update Allowed	Determines whether the item allows updates
Update Only If NULL	Determines whether the item allows updates only when the current value is NULL (This property applies to existing records only.)
Lock Record	Determines whether the record is locked when the item is modified (This property applies only to nonbase table items.)

Note: When you create an item in a data block, Form Builder assumes the item is a data item and sets its Database Item property to Yes. Data items are automatically included in any SELECT, UPDATE, and INSERT statements issued to the database. If an item you are creating is a control item, you must explicitly set its Database Item property to No.

Functional Properties



Order

Order Id 100 | Date Ordered 31-aug-1992

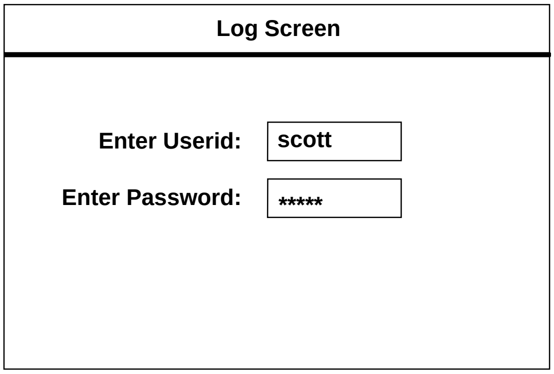
Date Shipped 10-sep-1992 | Payment Type CREDIT

Item

Id	Product Id	Price	Shipped	Item Total
1	10011	135	500	67,500.00
2	10013	380	400	152,000.00

Copyright © Oracle Corporation, 2000. All rights reserved.

Conceal Data Property



Log Screen

Enter Userid: scott

Enter Password: *****

Copyright © Oracle Corporation, 2000. All rights reserved.

Adding Functionality to a Text Item

Add to the default functionality of a text item by introducing some of the additional features:

Functional Property	**Function**
Enabled	Determines whether you can navigate to an item and manipulate it with the mouse (When this property is set to No, the item is disabled and appears grayed out on most window managers.)
Justification	Determines the text justification of the value within the item
Multi-Line	Determines whether the item can store multiple lines of text (Text items can be multiline only if they represent a VARCHAR2, or long base table column.)
Wrap Style	Determines whether a multiline text item wraps onto the next line; choose from None, Character, or Word
Case Restriction	Forces the case of the user’s input to Upper, Lower, or Mixed
Conceal Data	Determines whether the value in the item is echoed to the screen and hence visible to the user (This setting is typically used for password protection.)
Keep Cursor Position	Retains the cursor position so that when you reenter an item, it is in the same place
Automatic Skip	Determines whether the cursor automatically skips into the next item when this one is full (Auto Skip is used in conjunction with the Fixed Length property.)
Popup Menu	Specifies the pop-up menu within the current form module to display for the item

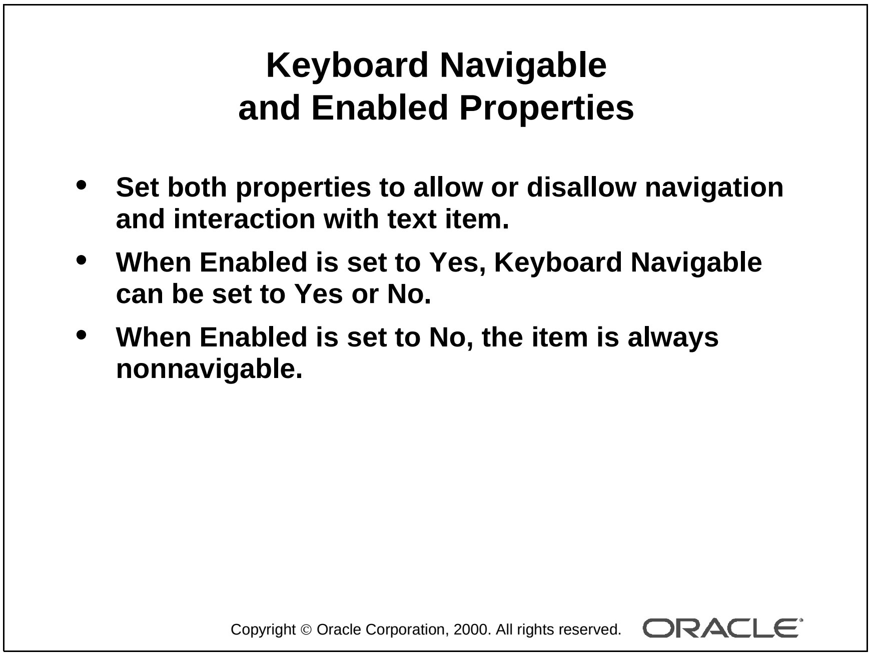
Note: The Enabled property set to No grays out the item. If you want the item to appear normally but do not want the users to change it, do the following:

- Set Insert Allowed to No.
- Set Update Allowed to No.
- Set Enabled to Yes.

Note: A pop-up menu is a context-sensitive menu that enables users to access common functions and commands quickly. It is a top-level object in the Object Navigator and belongs to a form module.

Technical Note

A pop-up menu belongs to the form module, as opposed to a form menu, which belongs to a separate menu module.



Keyboard Navigable and Enabled Properties

- Set both properties to allow or disallow navigation and interaction with text item.
- When Enabled is set to Yes, Keyboard Navigable can be set to Yes or No.
- When Enabled is set to No, the item is always nonnavigable.

Copyright © Oracle Corporation, 2000. All rights reserved.

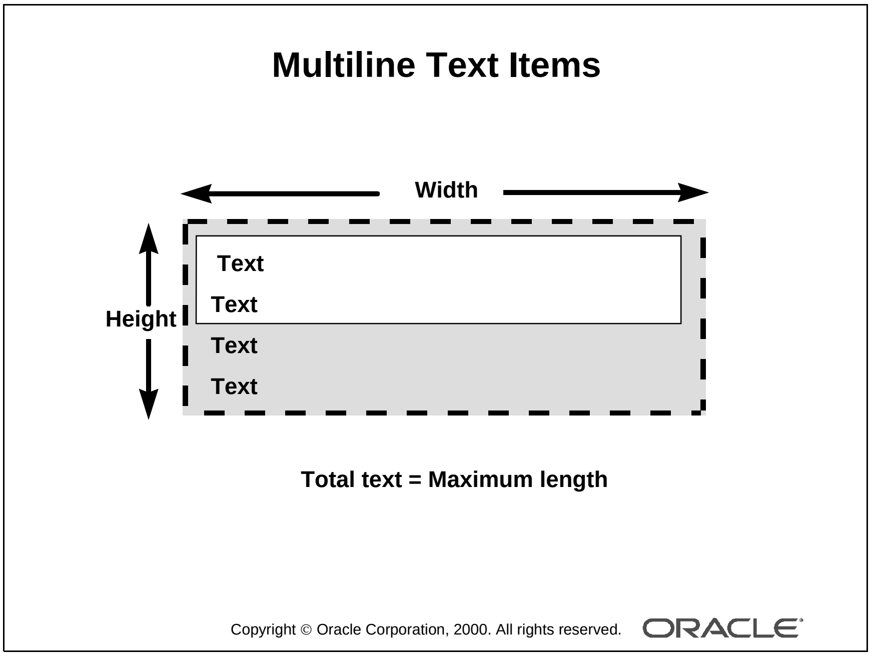
Instructor Note

Demonstration Using the `ordwk05.fmb` file, show the run-time effects of changing the Enabled, Justification, Multi-Line, Case Restriction, Conceal Data, and Automatic Skip properties.

Setting Keyboard Navigable and Enabled Properties

You can set the Keyboard Navigable and Enabled properties for items to specify whether operators can navigate to and interact with them.

Enabled	Keyboard Navigable	Navigation Behavior
Yes	Yes	Item is included during default navigation. The item can be navigated to and manipulated with the mouse.
Yes	No	Item is excluded during default navigation. The item can be navigated to and manipulated with the mouse.
No	No	Item is excluded during default navigation. The item cannot be navigated to or manipulated with the mouse.
No	Yes	Item is excluded during default navigation. The item cannot be navigated to or manipulated with the mouse. The Keyboard Navigable property is also effectively set to No.



Multiline Text Items

Width

Height

Text

Text

Text

Text

Total text = Maximum length

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Multiline Text Item

Use multiline text items to display the following:

- Addresses
- Comments
- Descriptions

The data in a multiline text item must be one of the following types:

- Char
- Alpha
- Long

When creating a multiline text item, consider the following properties:

- Width
- Height
- Font Size
- Maximum Length
- Show Vertical Scroll Bar

To see the desired number of lines, you need to alter the default values for the aforementioned properties; otherwise, the text item merely has the ability to store multilined data.

The number of characters that can fit on one line is dictated by the width of the text item and the font in use.

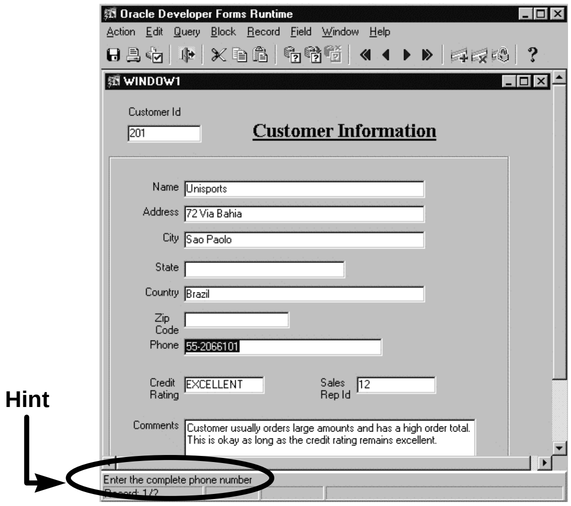
Setting the Justification

You can specify how text values are aligned in a text item by setting the Justification property to Left, Right, Center, Start, or End.

Note: Some window managers do not display right-aligned or centered text.

Setting right or center alignment for scrollable text items may result in values being hidden from the user.

Help Properties



Hint
Oracle Developer Forms Runtime
Action Edit Query Block Record Field Window Help
WINDOW1
Customer Information

Customer Id: 201

Name: Unisports

Address: 72 Via Bahia

City: Sao Paolo

State:

Country: Brazil

Zip Code:

Phone: 55-2066101

Credit Rating: EXCELLENT

Sales Rep Id: 12

Comments: Customer usually orders large amounts and has a high order total. This is okay as long as the credit rating remains excellent.

Enter the complete phone number

Record: 1/2

Copyright © Oracle Corporation, 2000. All rights reserved.

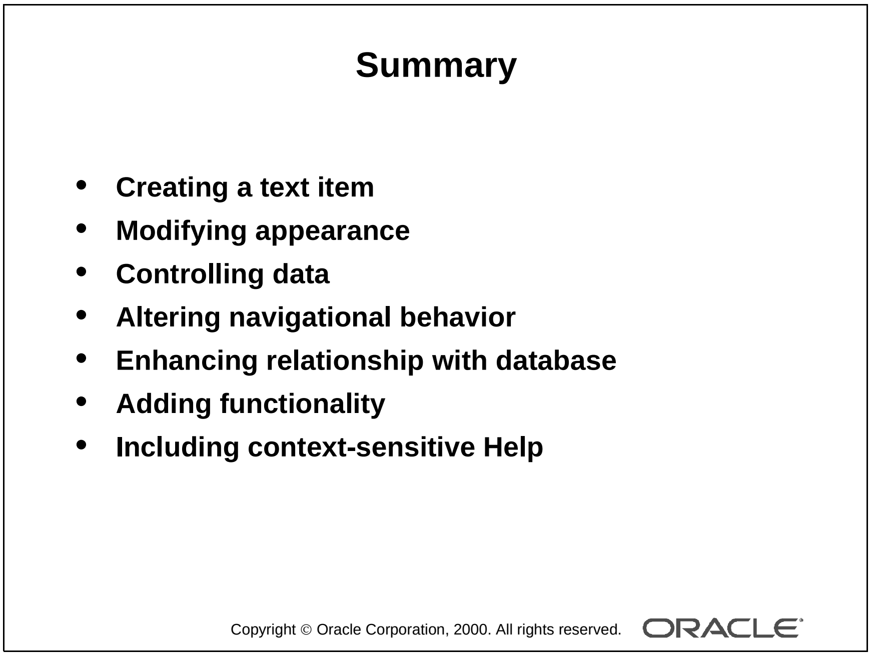
Instructor Note

Tool tips are discussed in Lesson 9.

Including Helpful Messages

Use the Help group properties to provide context-sensitive help to your user:

Help Property	**Function**
Hint	Writes item-specific Help text that is displayed on the message line at run time (The Help text is available when input focus is on the item.)
Display Hint Automatically	Determines whether the hint for the item is displayed automatically



Summary

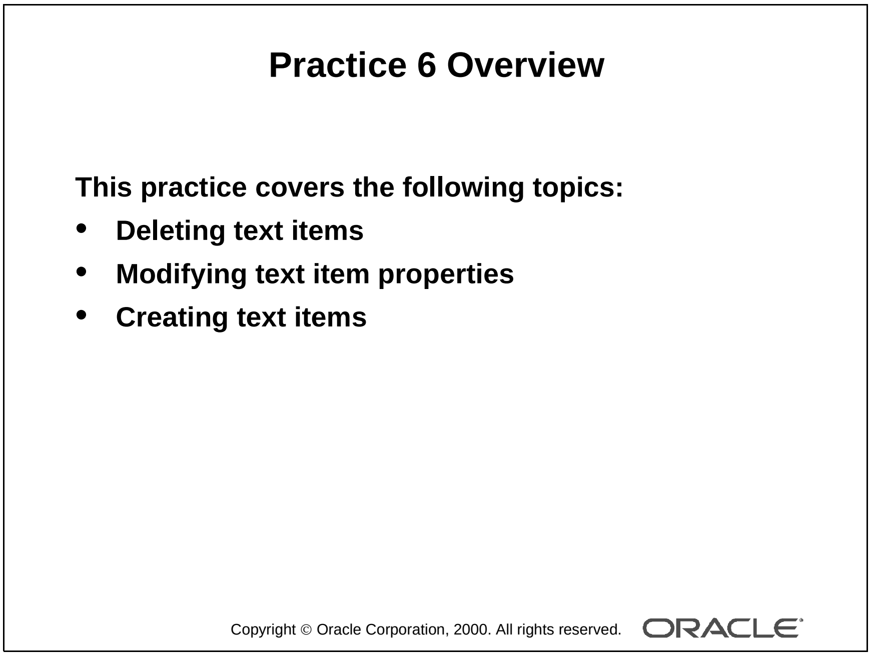
- Creating a text item
- Modifying appearance
- Controlling data
- Altering navigational behavior
- Enhancing relationship with database
- Adding functionality
- Including context-sensitive Help

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

This lesson showed you how to create and modify a text item that Form Builder creates for each column flagged for inclusion in a data block. In particular, text items have properties that enable you to do the following:

- Modify their appearance
- Control the data stored in the item
- Alter navigational behavior
- Enhance the relationship with the database
- Add functionality
- Include Help information



Practice 6 Overview

This practice covers the following topics:

- **Deleting text items**
- **Modifying text item properties**
- **Creating text items**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 6 in Appendix A, “Practice Solutions.”

Practice 6 Overview

In this practice session you will create text items, alter the behavior and the appearance of text items, and delete text items.

- Delete the region ID item in the CUSTOMERS form.
- Using the Property Palette, change the properties of several text items in the S_CUSTOMER data block to change their run-time appearance. Save and run the form after the changes are applied.
- In the ORDERS form, create new text items to hold the customer name and sales rep name values in the S_ORD block, and set the suggested properties. Change additional text item properties in the S_ORD, S_ITEM, and S_INVENTORY data blocks to change their run-time appearance and behavior. Save and run the form after the changes are applied.

Practice 6

CUSTG*XX* Form

1 Remove the Region ID item.

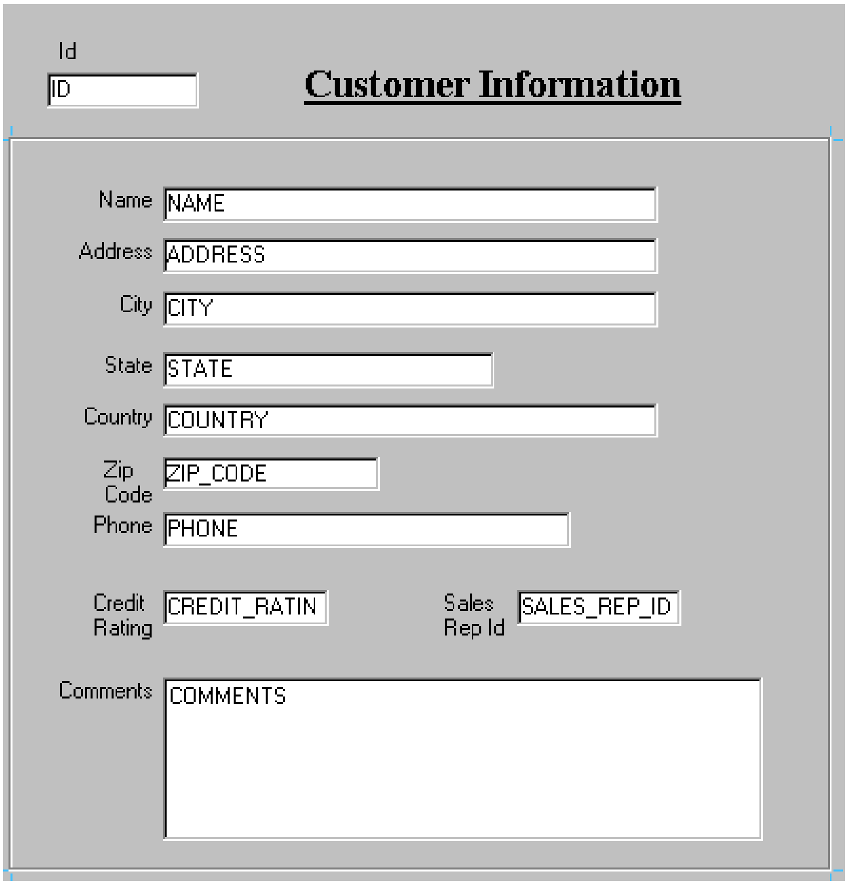
2 Make sure that the Comments item allows multiline text to display.

3 Automatically display a unique, new customer number for each new record and ensure that it cannot be changed. Use the S_CUSTOMER_ID sequence.

Note: This is not the only way to do this. Other methods are shown in later lessons.

4 In the CUSTG*XX* form, resize and reposition the items. Reorder the items in the Object Navigator. Use the screenshot and the table below as a guide. Resize items by setting the width according to the following property table.

Item	**Suggested Width**
ID	60
NAME	195
ADDRESS	195
CITY	195
STATE	130
COUNTRY	195
ZIP_CODE	85
PHONE	160
CREDIT_RATING	65
SALES_REP_ID	65
COMMENTS	236



Customer Information

Id	ID		
Name	NAME		
Address	ADDRESS		
City	CITY		
State	STATE		
Country	COUNTRY		
Zip Code	ZIP_CODE		
Phone	PHONE		
Credit Rating	CREDIT_RATIN	Sales Rep Id	SALES_REP_ID
Comments	COMMENTS		

5 Save and compile your form.
Deploy the form on the Web to test the changes.

ORDG*XX* Form

6 In the S_ORD block, create a new text item called Customer_Name.

Ensure that Customer_Name is not associated with the S_ORD table.

Do not allow insert, update, or query operations on this item, and make sure that navigation is possible only by means of the mouse. Set the Prompt text to Customer Name. Display this item on CV_ORDER canvas.

7 In the S_ORD block, create a new text item called Sales_Rep_Name.

Ensure that Sales_Rep_Name is not associated with the S_ORD table.

Do not allow insert, update, or query operations on this item and make sure that navigation is possible only by means of the mouse. Set the Prompt text to Sales Rep Name. Display this item on the CV_ORDER canvas.

8 Set the relevant property for Date_Ordered, so that it displays the current date whenever a new record is entered.

9 In the S_ITEM block, create a new text item called Item_Total.

Ensure that Item_Total is not associated with the S_ITEM table.

Do not allow insert, update, or query operations on this item and make sure that navigation is possible only by means of the mouse.

Allow numeric data only and display it by using a format of 999G990D99.

Set the Prompt text to Item Total. Display this item on the CV_ORDER canvas.

10 Justify the values of Price, Quantity, and Quantity_Shipped to the right.

11 Alter the Quantity_Shipped item, so that navigation is possible only by means of the mouse and updates are not allowed.

12 In the ORDG*XX* form, resize and reposition the items according to the screenshot and the table below.

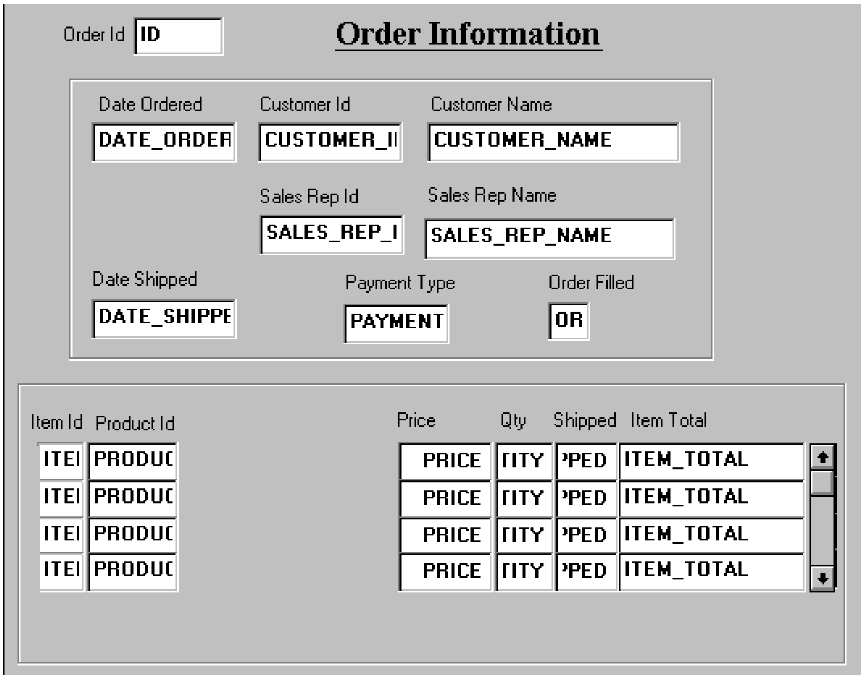
Resize the items by setting the width in the corresponding property palette. Drag and drop the items to reposition:

S_ORD Block Items	**Suggested Width**
ID	40
DATE_ORDERED	66
CUSTOMER_ID	66
CUSTOMER_NAME	116
SALES_REP_ID	66
SALES_REP_NAME	116
DATE_SHIPPED	66
PAYMENT_TYPE	48
ORDER_FILLED	18

S_ITEM Block Items	**Suggested Width**
ITEM_ID	20
PRODUCT_ID	40
PRICE	42
QUANTITY	26
QUANTITY_SHIPPED	26
ITEM_TOTAL	86

Instructor Note

Customer_Name and Sales_Rep_Name are created as text items in this course. They could also be created as display items. Display items use less memory than text items.



Order Id ID

Order Information

Date Ordered DATE_ORDER

Customer Id CUSTOMER_II

Customer Name CUSTOMER_NAME

Sales Rep Id SALES_REP_I

Sales Rep Name SALES_REP_NAME

Date Shipped DATE_SHIPPE

Payment Type PAYMENT

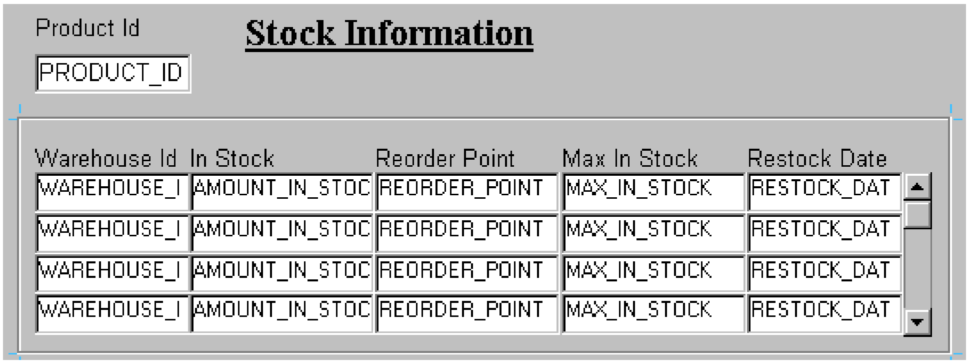
Order Filled OR

Item Id	Product Id	Price	Qty	Shipped	Item Total
ITEI	PRODUC	PRICE	TITY	PED	ITEM_TOTAL
ITEI	PRODUC	PRICE	TITY	PED	ITEM_TOTAL
ITEI	PRODUC	PRICE	TITY	PED	ITEM_TOTAL
ITEI	PRODUC	PRICE	TITY	PED	ITEM_TOTAL

13 In the S_INVENTORY block, alter the number of instances of the Product_ID so that it is displayed just once.

14 Arrange the items and boilerplate on CV_INVENTORY, so that it resembles the screenshot.

Hint: Set the Update Layout property for the frame to Manually.



Product Id

PRODUCT_ID

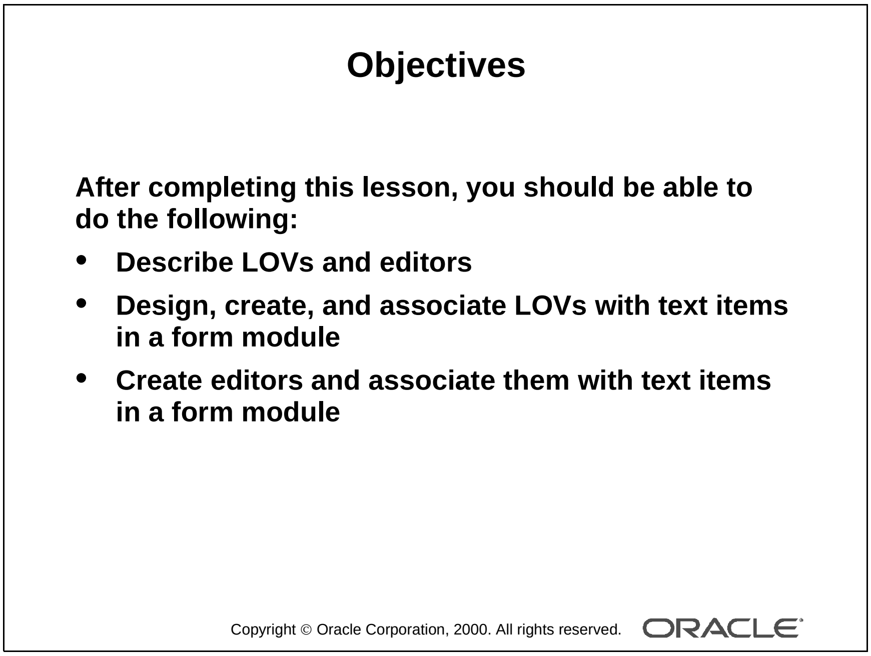
Stock Information

Warehouse Id	In Stock	Reorder Point	Max In Stock	Restock Date
WAREHOUSE_I	AMOUNT_IN_STOC	REORDER_POINT	MAX_IN_STOCK	RESTOCK_DAT
WAREHOUSE_I	AMOUNT_IN_STOC	REORDER_POINT	MAX_IN_STOCK	RESTOCK_DAT
WAREHOUSE_I	AMOUNT_IN_STOC	REORDER_POINT	MAX_IN_STOCK	RESTOCK_DAT
WAREHOUSE_I	AMOUNT_IN_STOC	REORDER_POINT	MAX_IN_STOCK	RESTOCK_DAT

15 Save, compile, and run the form to test the changes.

7

Creating LOVs and Editors



Objectives

After completing this lesson, you should be able to do the following:

- **Describe LOVs and editors**
- **Design, create, and associate LOVs with text items in a form module**
- **Create editors and associate them with text items in a form module**

Copyright © Oracle Corporation, 2000. All rights reserved.

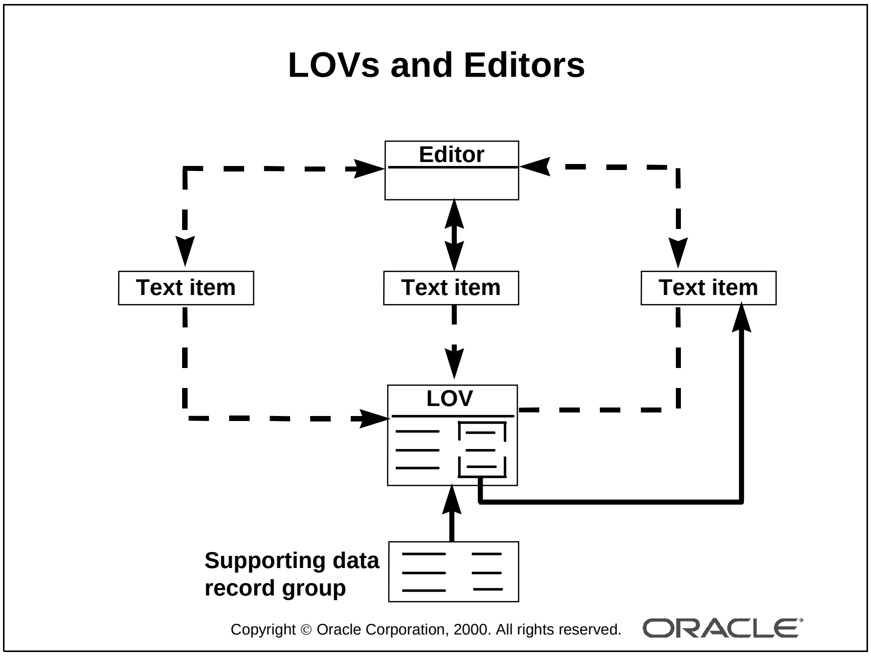
Schedule

Topic	**Timing**
Lecture	50 minutes
Practice	30 minutes
Total	80 minutes

Introduction

Overview

With Oracle Forms Developer you can enhance your application with lists of available values and text editors to supplement the text item object. In this lesson you will learn how to create lists of values (LOVs) and text editors, and to associate them with items in your application.



LOVs and Editors

Editor

Text item

Text item

Text item

LOV

Supporting data record group

Copyright © Oracle Corporation, 2000. All rights reserved.

What Are LOVs and Editors?

Lists of values (LOV) and *editors* are objects in a form module that each open their own window when activated at run time. They are defined at the form level, which means you can use them to support text items in any block of the form module.

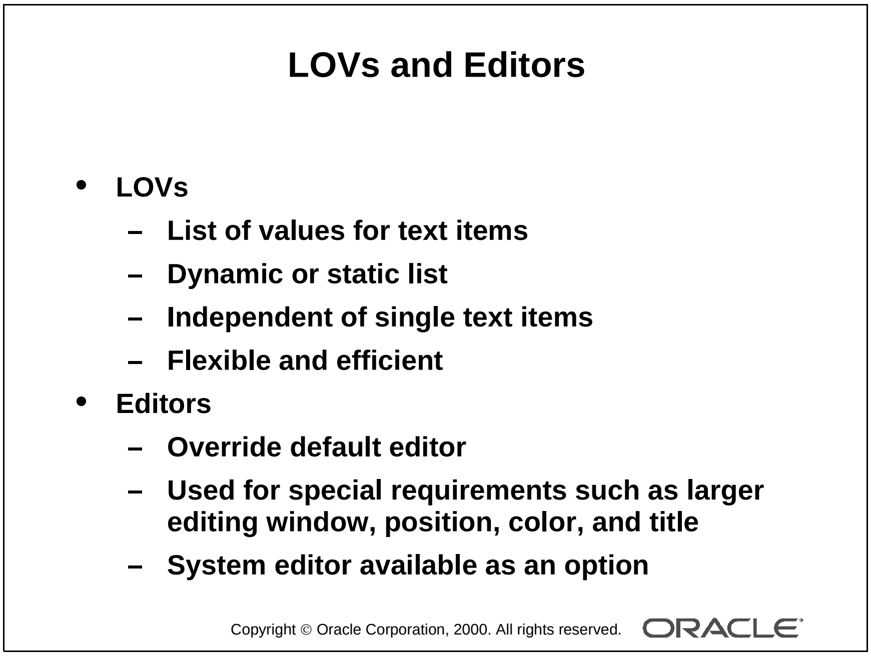
LOVs

An LOV is a scrollable pop-up window that provides a user with a simple mechanism to pick the value of an item from a multicolumn dynamic list. The user can reduce the lines displayed in the list by simple automatic reduction techniques, or by search strings.

Each line in an LOV can present several field values, with column headings above. You can design your LOV to retrieve some or all of the field values from the line chosen by the user, and place them into form items or variables.

LOVs have the following qualities:

- Dynamic: The list entries can change to reflect changes in the source data.
- Independent: The designer can invoke an LOV from any text item, or from outside a text item if called programmatically.
- Flexible: You can use the same LOV to support several items, if appropriate (for example, product_ID, product_name).
- Efficient: You can design LOVs to reuse data already loaded into the form, instead of accessing the database for every call. This is useful where data is relatively static.



LOVs and Editors

- **LOVs**
  - **List of values for text items**
  - **Dynamic or static list**
  - **Independent of single text items**
  - **Flexible and efficient**
- **Editors**
  - **Override default editor**
  - **Used for special requirements such as larger editing window, position, color, and title**
  - **System editor available as an option**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

- Pressing the [Delete Backward] key expands the reduced list of choices in the LOV. To find a key to match Delete Backward, select Help—>Keys to display Keys list.
- The EDIT_TEXTITEM built-in invokes the editor associated with the current text item. The SHOW_EDITOR built-in invokes a user-named editor at the specified display coordinates.

How to Use an LOV at Run Time

When a text item has an LOV attached, the List of Values lamp displays on the status line, while the cursor is in the item.

1. Either press the List of Values key, or select Edit—>Display List to invoke the LOV.
2. Select an entry in the displayed list. You can type characters to automatically reduce the list, or enter a search string in the Find field.
3. Click OK to retrieve the line value.

Note: Automatic reduction works by comparing the search string entered with the values displayed in the first column of the LOV. If you start your search criteria with a % symbol, Form Builder performs a search on *all* LOV columns.

Editors

With a text editor enabled the user can view multiple lines of a text item simultaneously, search and replace text in it, and generally modify the value of an item from this separate window.

You can use one of three editors at run time:

- Form Builder default editor
- User-named editor
- System editor

Every text item has the default editor available, but you can design your own replacement editor for those items that have special requirements such as larger editing window, position, color, and title.

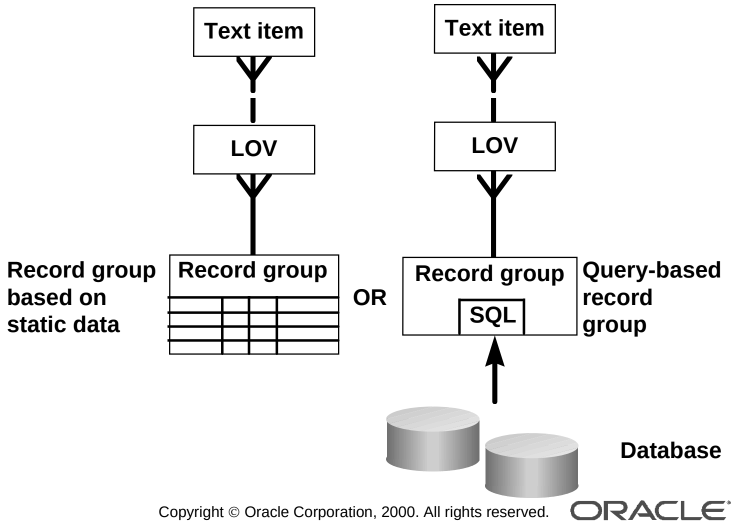
By overriding the default editor for a text item, you can provide a larger editing window for items with potentially large textual values. Optionally, use an external system editor.

How to Use an Editor at Run Time

With the cursor in the text item to be edited, follow these steps:

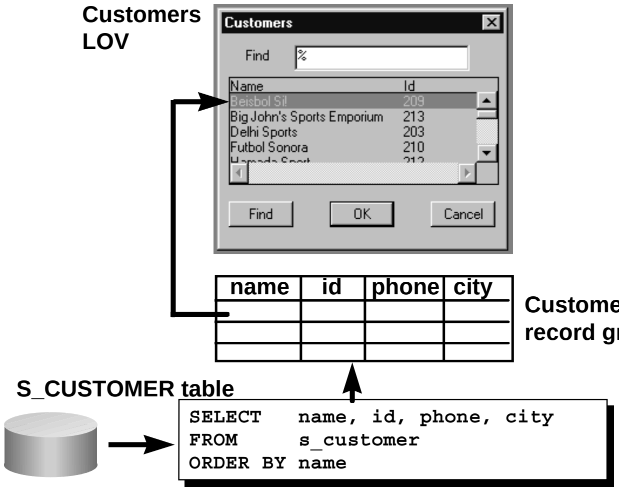
1. Press the Edit key, or select Edit—>Edit to invoke the attached editor.
2. Edit the text in the Editor window. Form Builder editors provide a Search button that invokes an additional search-and-replace dialog box for manipulating text.
3. Click OK to write your changes back to the text item.

LOVs and Record Groups



Copyright © Oracle Corporation, 2000. All rights reserved.

LOVs



Name	Id
Beisbol Si!	209
Big John's Sports Emporium	213
Delhi Sports	203
Futbol Sonora	210
Hamada Sport	212

```
SELECT   name, id, phone, city
FROM     s_customer
ORDER BY name
```

Copyright © Oracle Corporation, 2000. All rights reserved.

Defining an LOV

Designing an LOV

When you build an LOV, consider the following objects:

- Record group: A Form Builder object that is used to store the array of values that are presented by an LOV (The record group can be created first or as part of the LOV creation process if based on a query.)
- LOV: The list itself, which presents *one or more* column values from the supporting record group in the LOV window (It enables the user to select values, and then write values back to specified items or variables.)

  Text items: The main text item that you attach to an LOV is usually one that the LOV returns a value to. You can call the LOV from this item to provide possible values for it. A single LOV can return values to several items, and you may want to attach it to these items as well in your application.

  In fact, you can attach the LOV to any text item from which the same list of values needs to be viewed, whether or not it will receive a value.

Record Groups

A *record group* is a column-and-row structure stored within Forms Runtime memory and is similar to the structure of a database table. It holds records that can be reused by other Oracle Forms Developer applications and Oracle Reports Developer applications, hence reducing repeated access to external data.

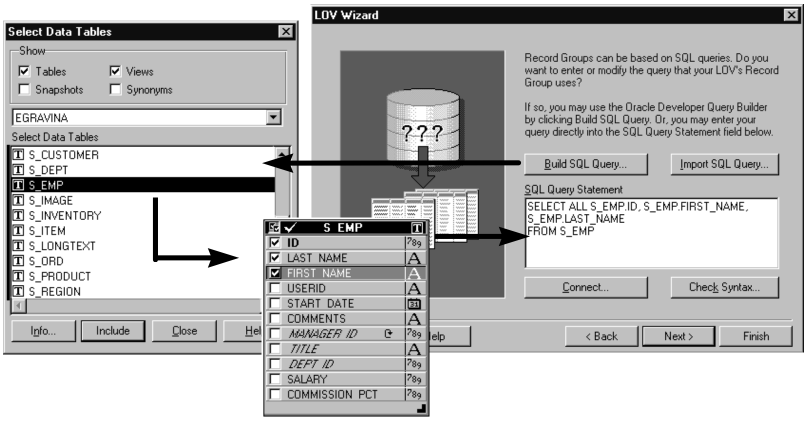
Record groups can be designed to contain static values. Alternatively, they can be populated programmatically at run time or, most commonly, populated by a SQL query. In this lesson, you use record groups to support LOVs.

Record groups can provide the following:

- Data that is presented by LOVs
- Data for dynamic list items
- Data to be passed to Report Builder and Graphics Builder
- Other application-defined uses

Note: Because LOVs and record groups are separate objects, you can create multiple LOVs based on the same record group.

Creating an LOV Using the LOV Wizard: SQL Query Page



Show

☑ Tables ☑ Views

☐ Snapshots ☐ Synonyms

EGRAVINA

Select Data Tables

- S_CUSTOMER
- S_DEPT
- S_EMP
- S_IMAGE
- S_INVENTORY
- S_ITEM
- S_LONGTEXT
- S_ORD
- S_PRODUCT
- S_REGION

S_EMP

☑ ID

☑ LAST_NAME

☑ FIRST_NAME

☐ USERID

☐ START_DATE

☐ COMMENTS

☐ MANAGER_ID

☐ TITLE

☐ DEPT_ID

☐ SALARY

☐ COMMISSION_PCT

Record Groups can be based on SQL queries. Do you want to enter or modify the query that your LOV's Record Group uses?

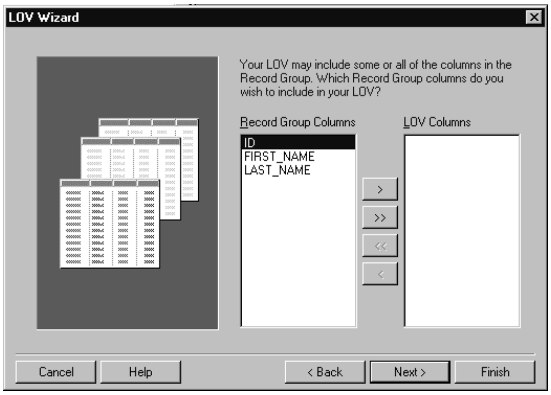
If so, you may use the Oracle Developer Query Builder by clicking Build SQL Query. Or, you may enter your query directly into the SQL Query Statement field below.

SQL Query Statement

```
SELECT ALL S_EMP.ID, S_EMP.FIRST_NAME,
S_EMP.LAST_NAME
FROM S_EMP
```

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating an LOV Using the LOV Wizard: Column Selection Page



Your LOV may include some or all of the columns in the Record Group. Which Record Group columns do you wish to include in your LOV?

Record Group Columns

- ID
- FIRST_NAME
- LAST_NAME

LOV Columns

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating an LOV by Using the LOV Wizard

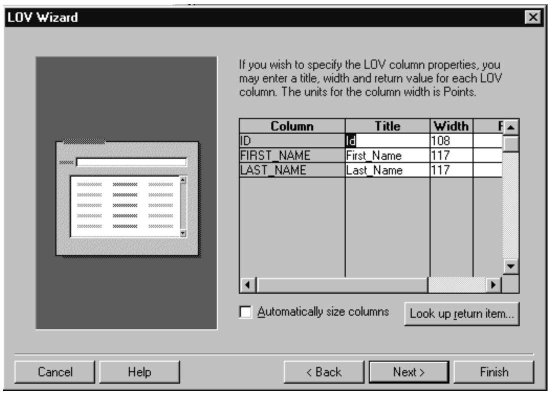
1. Launch the LOV Wizard.

   The Welcome page is displayed. Click Next.

2. Specify the LOV source in the LOV Source page. Choose an existing record group or create a new one based on a query. The New Record Group based on a query radio button is set by default. Click Next to select the default.
3. In the SQL Query page specify the query used to construct the record group.
   - Click the Build SQL Query button to use the Query Builder.
   - Click the Import SQL Query button to import a query from a file.
   - To enter the query directly, type the SQL syntax in the SQL Query Statement field. Then click the Check Syntax button.
4. In the Column Selection page, select the record columns that you want to include in the LOV.

Note: See Appendix G for more information about Query Builder.

Creating an LOV Using the LOV Wizard: Column Properties Page



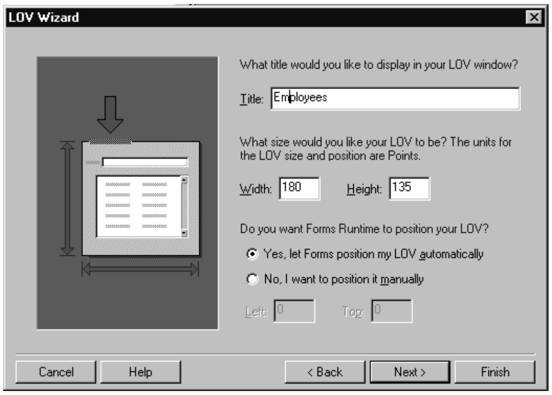
If you wish to specify the LOV column properties, you may enter a title, width and return value for each LOV column. The units for the column width is Points.

Column	Title	Width	F
ID	Id	108	
FIRST_NAME	First_Name	117	
LAST_NAME	Last_Name	117	

Copyright © Oracle Corporation, 2000. All rights reserved.

ORACLE®

Creating an LOV Using the LOV Wizard: Display Page



What title would you like to display in your LOV window?

Title: Employees

What size would you like your LOV to be? The units for the LOV size and position are Points.

Width: 180 Height: 135

Do you want Forms Runtime to position your LOV?

● Yes, let Forms position my LOV automatically

○ No, I want to position it manually

Left: 0 Top: 0

Copyright © Oracle Corporation, 2000. All rights reserved.

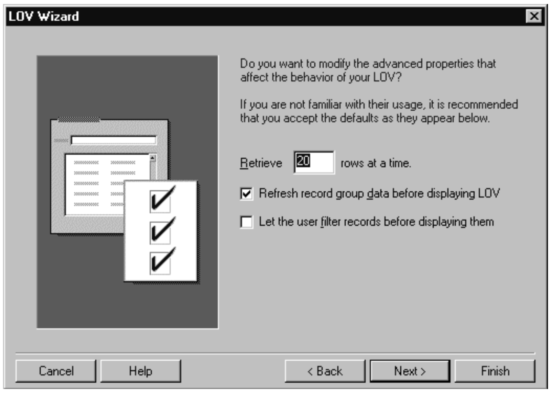
ORACLE®

Creating an LOV by Using the LOV Wizard (continued)

5 In the Column Properties page, specify the title, width and return value for each LOV column. Note that the Return Value Into item is optional.

6 In the LOV Display page, specify the title, width, and height of the LOV window.

Creating an LOV Using the LOV Wizard: Advanced Properties Page



LOV Wizard

Do you want to modify the advanced properties that affect the behavior of your LOV?

If you are not familiar with their usage, it is recommended that you accept the defaults as they appear below.

Retrieve 20 rows at a time.

☑ Refresh record group data before displaying LOV

☐ Let the user filter records before displaying them

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

As in earlier releases of Oracle Forms Developer, an LOV can be created manually. Demonstrate this for the students.

- Select the LOV node in the Object Navigator and click the Create icon.
- Select Build a new LOV manually. A new LOV appears under the LOV node in the Object Navigator.
- Invoke the property palette for the new LOV to set the required properties.

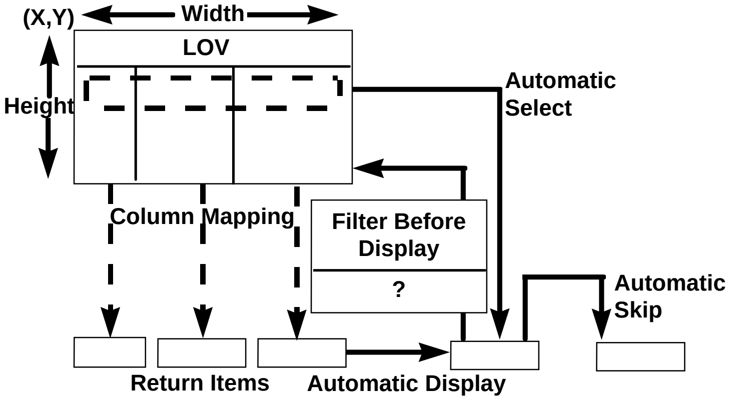
Creating an LOV by Using the LOV Wizard (continued)

7 In the Advanced Options page, set the additional advanced properties. Specify:

- The number of records to be fetched from the database
- If the user should be presented with a dialog box to add criteria before the LOV is displayed
- If the LOV records should be queried each time the LOV is invoked

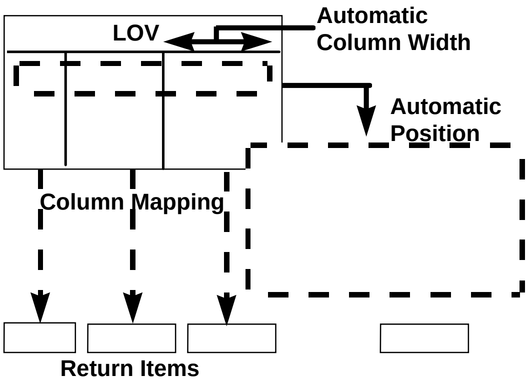
8 In the Finish page, click Finish to complete the LOV creation process.

LOV Properties


(X,Y)
Width
LOV
Height
Automatic Select
Filter Before Display
?
Column Mapping
Automatic Skip
Return Items
Automatic Display

Copyright © Oracle Corporation, 2000. All rights reserved.

LOV Properties


LOV
Automatic Column Width
Automatic Position
Column Mapping
Return Items

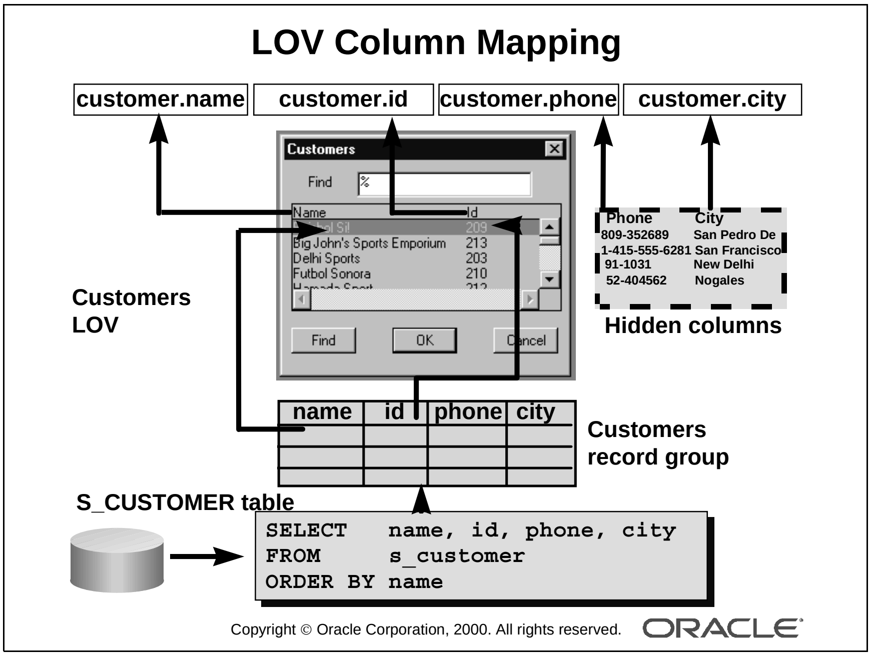
Copyright © Oracle Corporation, 2000. All rights reserved.

Setting LOV Properties

After you create an LOV, open its Property Palette to define its properties.

Property	Function
Title	Specifies a title for the LOV
X Position and Y Position	Screen coordinates for the LOV window in the current form coordinate units (Choose a position that is suitable for the items that the LOV supports.)
Width and Height	Size of the LOV window in the current form coordinate units (The user can adjust this, but choose a size that is suitable for the data.)
Column Mapping Properties (More . . .)	Opens the LOV Column Mapping window (discussed later in this lesson)
Filter Before Display	Determines whether users should be prompted with a dialog box that enables them to enter a search value before the LOV is invoked (This value is used as an additional restriction on the first column in the query.)
Automatic Display	Determines whether the LOV should be invoked automatically when the cursor enters an item to which the LOV is attached
Automatic Refresh	When this property is set to Yes, the record group reexecutes its query every time the LOV is invoked. When this property is set to No, the record group query fires only the first time the LOV is invoked within a user session. Subsequent LOV calls use current record group data (more efficient for data that will not change much.)
Automatic Select	Determines whether the LOV should close and return values automatically when reduced to a single entry
Automatic Skip	Determines whether the cursor skips to the next navigable item when the operator selects a value from the LOV to populate the text item
Automatic Position	Determines whether Form Builder automatically positions the LOV near the field from which it was invoked
Automatic Column Width	Determines whether Form Builder automatically sets the width of each column to display the entire column title when the column title width is longer than the column display width

Note: More than one LOV can be based on the same record group. When this is the case and you set Automatic Refresh to No, Form Builder will not reexecute the LOV query once any of the LOVs is invoked.



LOV Column Mapping

Name	Id
[illegible]	209
Big John's Sports Emporium	213
Delhi Sports	203
Futbol Sonora	210
Hamada Sport	212

Phone	City
809-352689	San Pedro De
1-415-555-6281	San Francisco
91-1031	New Delhi
52-404562	Nogales

```
SELECT   name, id, phone, city
FROM     s_customer
ORDER BY name
```

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

- Do not put a colon in front of the Return Item object names.
- To add extra spaces between columns in the LOV window, set the Display Width wider than the column default width. However, you cannot increase the width between a number column and a nonnumber column this way because LOVs display numbers right-justified.
- You can also attach an LOV to a text item programmatically with the SET_ITEM_PROPERTY built-in.

The Column Mapping Properties

When you click the More property control button for Column Mapping Properties, the LOV Column Mapping dialog box opens.

Column Mapping Element	**Description**
Column Names (List)	Lets you select an LOV column for mapping or defining a column

Return Item	Specifies the name of the form item or variable to which Form Builder should assign the column value.
	Use one of the following:
	`block_name.item_name`
	`GLOBAL.variable_name`
	`PARAMETER.parameter_name`
	If null, the column value is not returned from LOV.

Display Width	Width of column display in LOV (A 0 value causes the column to be hidden, although its value remains available for return.)
Column Title	Heading for column in LOV window

To set a column mapping in this dialog, first select the column from the Column Names list, then set the other mapping values, as required.

Note: The record group columns and LOV columns must remain compatible.

You can modify the record group query from its own properties list.

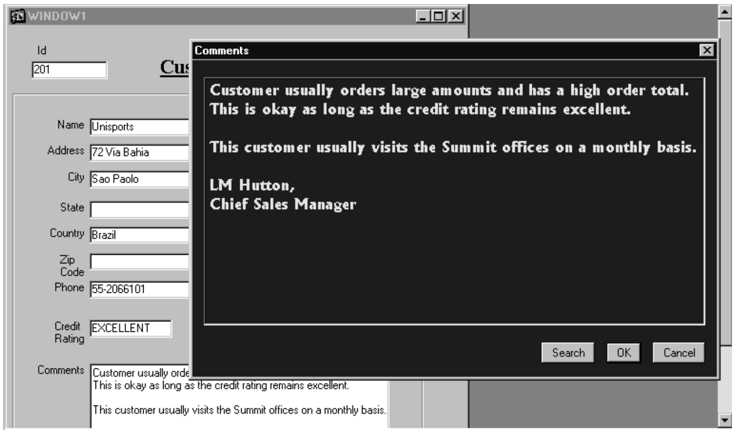
Associating an LOV with a Text Item

So that the user can invoke an LOV from a text item, you must specify the LOV name in the Property Palette of the text item.

1. Select the text item in the Object Navigator from which the LOV is to be accessible.
2. In the item Property Palette, set the List of Values property to the required LOV.

Remember that the List of Values lamp is displayed when the user navigates to this text item, indicating that the LOV is available through the List of Values key or menu command.

Editors



Id 201

Name Unisports

Address 72 Via Bahia

City Sao Paolo

State

Country Brazil

Zip Code

Phone 55-2066101

Credit Rating EXCELLENT

Comments Customer usually orde
This is okay as long as the credit rating remains excellent.

This customer usually visits the Summit offices on a monthly basis.

Customer usually orders large amounts and has a high order total.
This is okay as long as the credit rating remains excellent.

This customer usually visits the Summit offices on a monthly basis.

LM Hutton,
Chief Sales Manager

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Show how to create an editor.

Defining an Editor

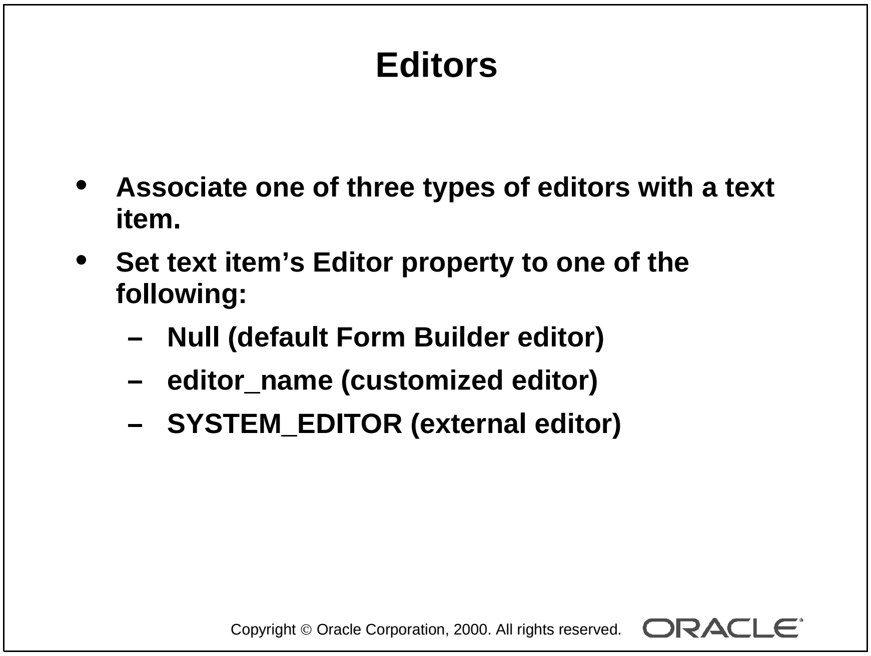
If the user needs to use an editor on text values, the default Form Builder editor is usually sufficient for most items. However, you can design your own customized editor as an object in a form module, and then attach it to the text items that need it.

How to Create a Customized Editor

1. Select the Editors node in the Object Navigator, then click Create. A new editor object is displayed in the list.
2. Select the new editor in the Object Navigator, and then access its Property Palette, where you can set its name and other properties.

The following properties show the individual tailoring that is possible by creating your own editor. Properties are abridged and summarized.

Property	**Function**
Title/Bottom Title	Displays at top or bottom of editor window
Width/Height	These properties control the size of the editor and hence its editing area
X Position/Y Position	Screen position for window; can also be defined by a text item property
Wrap Style	Specifies how text is displayed when a line of text exceeds the width of the editing area (Choose among None, Character, or Word.)
Show Vertical Scroll Bar	To add a vertical scroll bar to the editor, specify Yes for this property.



Editors

- **Associate one of three types of editors with a text item.**
- **Set text item’s Editor property to one of the following:**
  - **Null (default Form Builder editor)**
  - **editor_name (customized editor)**
  - **SYSTEM_EDITOR (external editor)**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

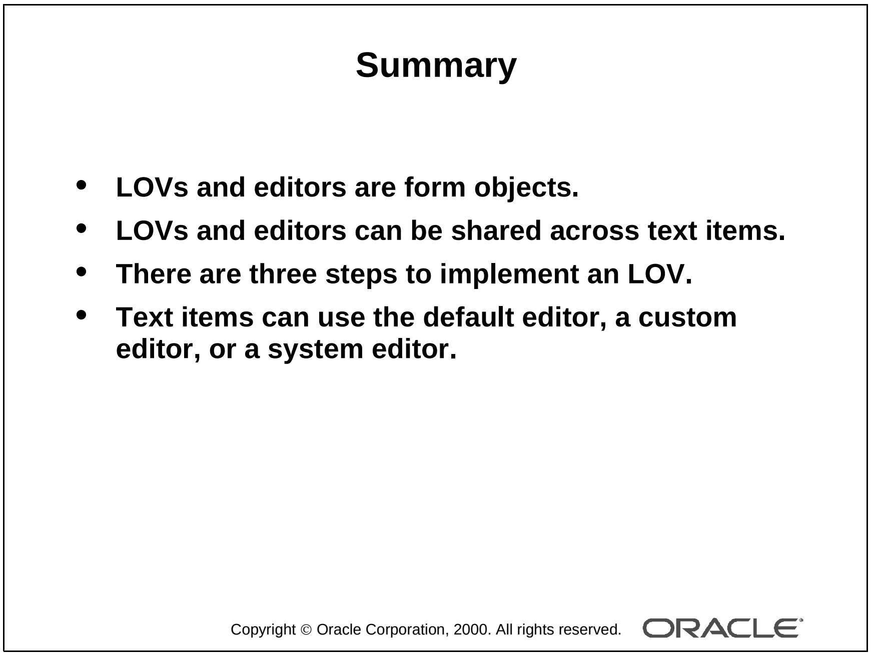
Associating an Editor with a Text Item

To associate an editor with a text item, you must specify the editor in the Property Palette of the text item.

Select the text item in the Object Navigator from which the editor is to be accessible.

In the item Property Palette, set the Editor property to one of the following settings:

- Null: The text item uses the default Form Builder editor.
- editor_name: The text item uses the named editor that you have created and customized in this module.
- SYSTEM_EDITOR: The item uses the external ASCII editor defined in the system editor environment variable (which varies according to platform).



Summary

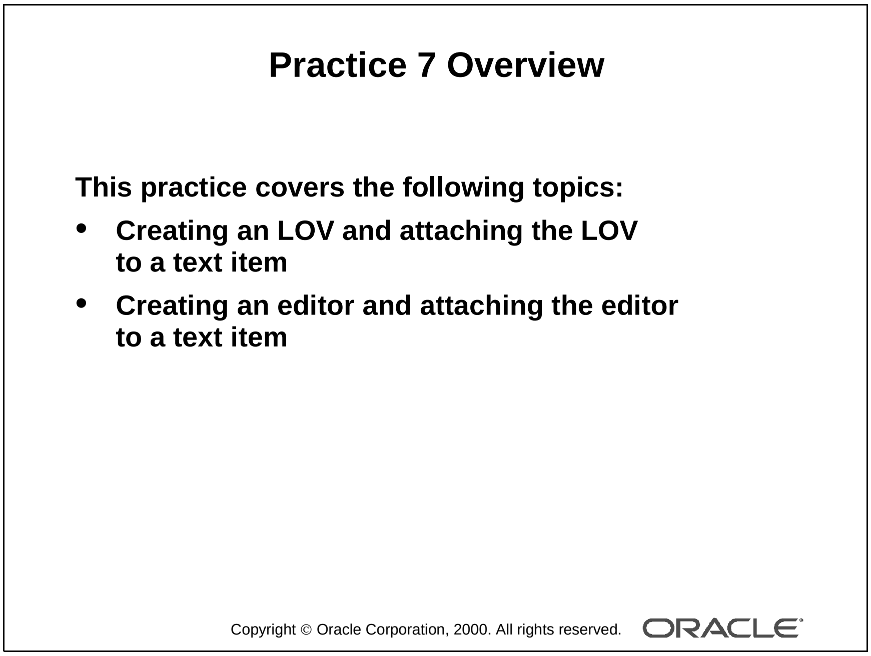
- LOVs and editors are form objects.
- LOVs and editors can be shared across text items.
- There are three steps to implement an LOV.
- Text items can use the default editor, a custom editor, or a system editor.

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

In the lesson you learned that lists of values (LOVs) and text editors can be used to support text items. Both LOVs and editors are objects in a form module that open their own window when activated at run time and are used to support text items in any block of the form module.

- LOVs and editors can be shared across text items.
- The steps to implement an LOV are:
  - **a** Create a new LOV (and record group).
  - **b** Define column mapping for return items.
  - **c** Attach the LOV to text items, as required.
- Text items can use the default editor, a user-named editor, or a system editor.



Practice 7 Overview

This practice covers the following topics:

- **Creating an LOV and attaching the LOV to a text item**
- **Creating an editor and attaching the editor to a text item**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 7 in Appendix A, “Practice Solutions.”

Practice 7 Overview

In this practice session, you will create two LOVs and an editor.

- Using the LOV Wizard, create an LOV in the ORDERS form to display product numbers and their descriptions. Attach the LOV to the Product_ID item in the S_ITEM data block. Save and run the form.
- Using the LOV wizard, create an LOV in the CUSTOMERS form to display sales representatives' numbers and their names. Attach the LOV to the Sales_Rep_ID item in the S_CUSTOMER data block. Save and run the form.
- Create an editor in the CUSTOMERS form, and attach it to the Comments item. Save and run the form.

Practice 7

1 In the ORDG*XX* form, create an LOV using the LOV Wizard to display product numbers and descriptions to be used with the Product_Id item in the S_ITEM block. Use the S_PRODUCT table, Id, and Name columns.

Assign a title of Products to the LOV. Assign a column width of 25 for ID, and assign the LOV width of 200 and a height of 250. Position the LOV 30 pixels below and to the right of the upper lefthand corner. For the ID column, set the return item to S_ITEM.PRODUCT_ID. Attach the LOV to the Product_Id item in the S_ITEM block. Change the name of the LOV to PRODUCTS_LOV.

2 Save and compile your form. Deploy the form on the Web to test the changes.

3 In the CUSTG*XX* form, create an LOV to display sales representatives' numbers and their names, using the LOV Wizard. Use the S_EMP table, Id, First_Name, and Last_Name columns. Concatenate the First_Name and the Last_Name columns and give an alias such as Name.

Assign a title of Sales Representatives to the LOV. Assign a column width of 20 for ID, and assign the LOV a width of 200, and a height of 250. Position the LOV 30 pixels below and to the right of the upper lefthand corner. For the ID column, set the return item to S_CUSTOMER.SALES_REP_ID. Attach the LOV to the Sales_Rep_Id item in the S_CUSTOMER block.

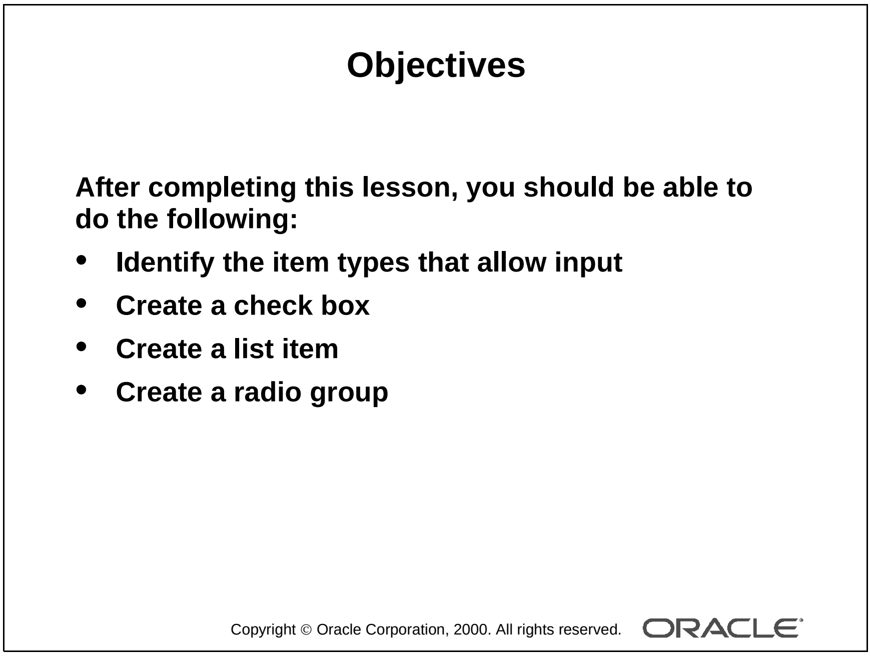
Change the name of the LOV to SALES_REP_LOV.

4 In the CUSTG*XX* form, create an editor and attach it to the Comments item. Set the title to Comments, the background color to gray, and the foreground color to yellow.

5 Save, compile, and run the form to test the changes. Resize the window if necessary.

8

Creating Additional Input Items



Objectives

After completing this lesson, you should be able to do the following:

- **Identify the item types that allow input**
- **Create a check box**
- **Create a list item**
- **Create a radio group**

Copyright © Oracle Corporation, 2000. All rights reserved.

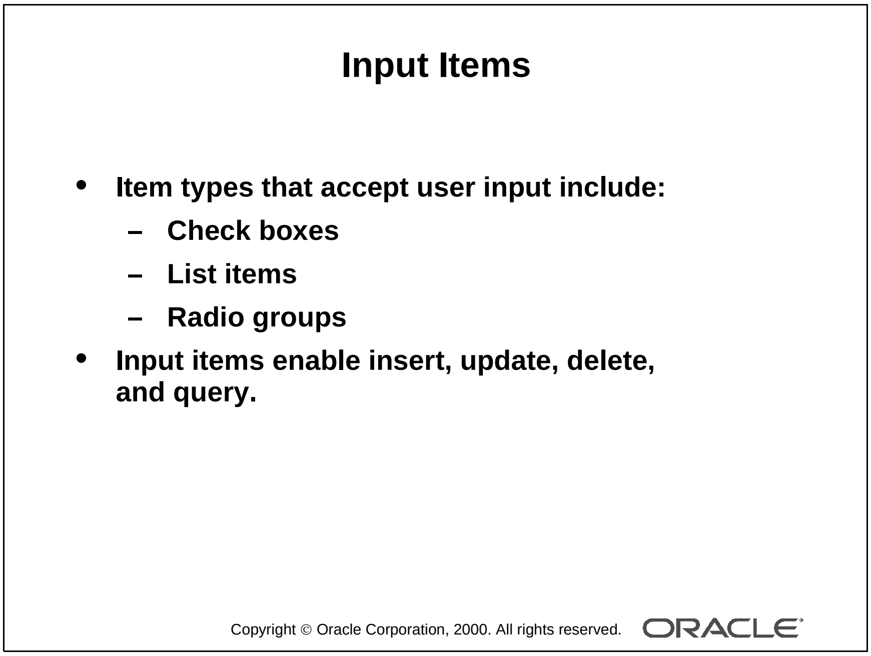
Schedule

Topic	**Timing**
Lecture	60 minutes
Practice	45 minutes
Total	105 minutes

Introduction

Overview

In addition to text items, Oracle Forms Developer provides a variety of other item types. These can be divided into two groups: those that accept input and those that do not. This lesson covers input items and how they are used.



Input Items

- Item types that accept user input include:
  - Check boxes
  - List items
  - Radio groups
- Input items enable insert, update, delete, and query.

Copyright © Oracle Corporation, 2000. All rights reserved.

What Are Input Items?

Input item is a generic term for Form Builder item types that accept user input.

These item types include the following:

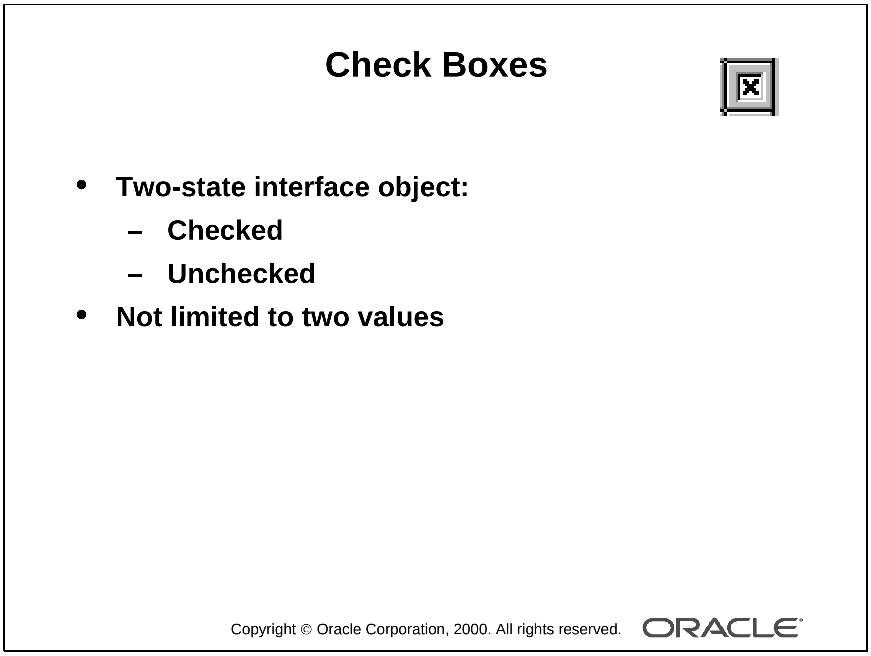
- Check box
- List item
- Radio group

What Can You Do with Input Items?

When you create input items, they already have some initial functionality. Through items you can interact with the database in the following ways:

- Insert values
- Update existing values
- Delete existing values
- Query existing values

Note: You can add functionality to input items with triggers and PL/SQL program units.



Check Boxes

- **Two-state interface object:**
  - **Checked**
  - **Unchecked**
- **Not limited to two values**

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Check Box

What Is a Check Box?

A *check box* is a two-state interface object that indicates whether a certain value is ON or OFF. The display state of a check box is always either checked or unchecked.

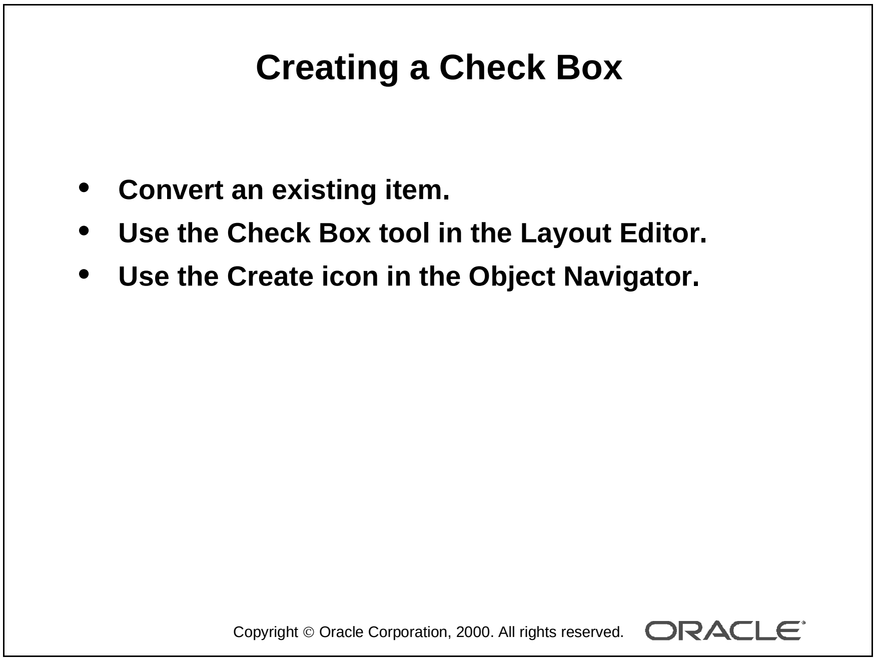
Although a check box is limited to two states, it is not limited to just two values. You specify the value to represent Checked, the value to represent Unchecked, and how other values are processed.

You can use check boxes to enhance the user interface by converting existing items that have two possible states.

Using a Check Box at Run Time

You can do the following at run time:

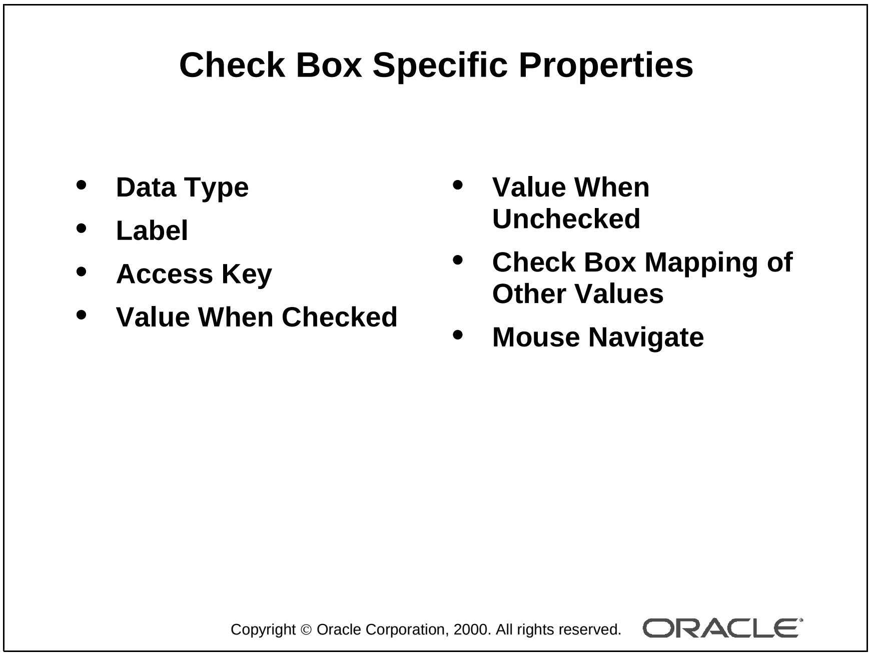
- Set check box values in the following ways:
  - By user input
  - By means of the Initial Value property
  - Programmatically
- Query checked values.
- Query unchecked values.
- Ignore check box values in Enter Query mode by disabling the item with [Shift] + Click.



Creating a Check Box

- Convert an existing item.
- Use the Check Box tool in the Layout Editor.
- Use the Create icon in the Object Navigator.

Copyright © Oracle Corporation, 2000. All rights reserved.



Check Box Specific Properties

- Data Type
- Label
- Access Key
- Value When Checked
- Value When Unchecked
- Check Box Mapping of Other Values
- Mouse Navigate

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Check Box

A check box can be created in three ways:

- Converting an existing item
- Using the Check Box tool in the Layout Editor
- Using the Create icon in the Object Navigator

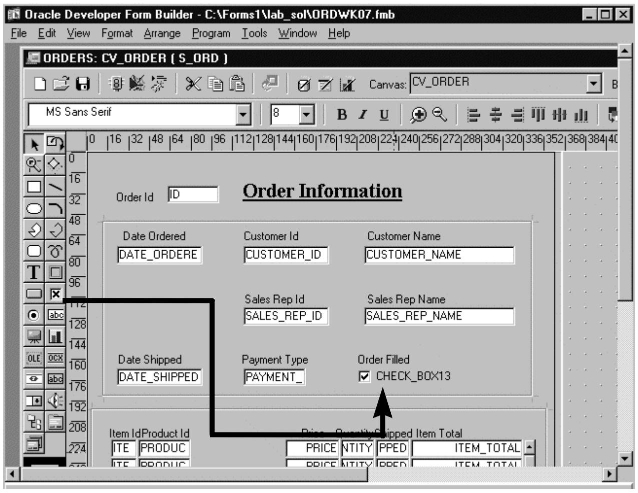
Item Properties Specific to the Check Box

Property	**Function**
Data Type	Determines the data type (There is a restricted choice of CHAR, NUMBER, and DATE.)
Label	Specifies the text label that is displayed adjacent to the check box item (This is independent of the check box value.)
Access Key	Determines which combination of keys can be used to navigate to this item and check or uncheck it
Initial Value	Specifies the initial value of the item and thus determines whether the check box is initially checked or unchecked
Value When Checked	Specifies a value to represent the checked state of the check box (The value must be compatible with the data type specified.)
Value When Unchecked	Specifies a value to represent the unchecked state of the check box (The value must be compatible with the data type specified.)
Check Box Mapping of Other Values	Determines how other values are processed (Choose from NOT ALLOWED, CHECKED, and UNCHECKED.)
Mouse Navigate	Determines whether Form Builder navigates to the item and moves input focus to it when the user activates the item with the mouse (The default setting is Yes.)

Instructor Note

The Mouse Navigate property applies only to GUI applications and is valid only for buttons, check boxes, list items, sound items, and radio group items. When Mouse Navigate is set to Yes, Form Builder navigates to the item, firing any appropriate navigation and validation triggers on the way.

Creating a Check Box



Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

Demonstration Using the `ordwk07.fmb` file, convert the Order_Filled item to a check box item. Point out that the object icon for this check box item changes automatically in the Object Navigator.

How to Convert an Existing Item into a Check Box

You can convert an existing item into a check box by changing the Item Type property to Check Box in the Property Palette and setting other relevant properties.

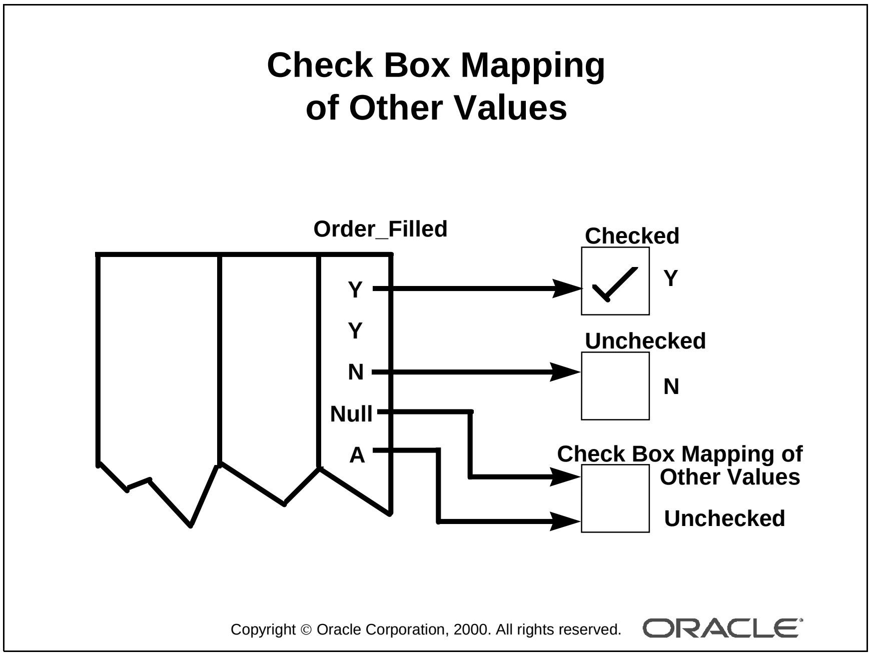
1. Invoke the Property Palette for the item that you want to convert.
2. Set the Item Type property to Check Box.
3. Enter a check box label.
4. Enter values for the checked and the unchecked states.
5. Set the Check Box Mapping of Other Values property.
6. Enter an initial value for the check box item.

Note: The check box label that you specify is displayed to the right of the check box element at run time. If the complete label name is not displayed, adjust it in the Layout Editor. If the item already has a prompt, delete it in the item Property Palette.

How to Create a Check Box in the Layout Editor

You can also create a check box by using the Check Box tool in the Layout Editor.

1. Invoke the Layout Editor.
2. Set the canvas and block to those on which you want the check box item to be displayed.
3. Click the Check Box tool.
4. Click the canvas in the position where you want the check box to be displayed.
5. Double-click the check box to invoke its Property Palette.
6. Set the properties as required.



Check Box Mapping of Other Values

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

You must specify a valid initial value, except under either of the following conditions:

- The Check Box Mapping of Other Values property is set to Checked or Unchecked.
- The value associated with Checked or Unchecked is NULL.

Dealing with Other Values

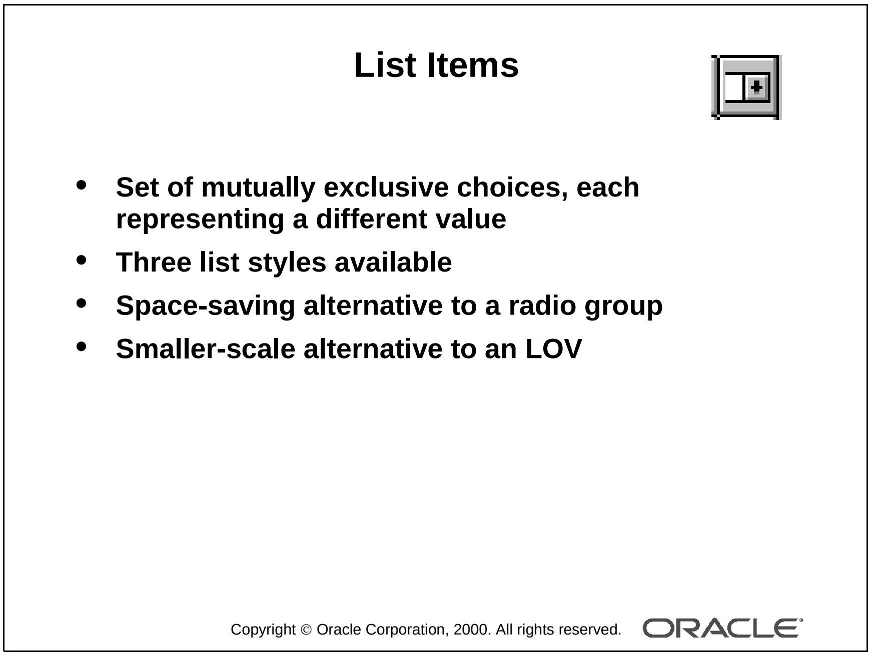
If your base table column accepts other values, then your check box should account for them. You can assign other values to either the checked or unchecked states by using the Check Box Mapping of Other Values property. Alternatively, you can choose not to accept other values with the Not Allowed setting.

Note: If you choose not to accept other values and they exist in the base table column, Form Builder ignores the entire record during query processing.

Dealing with Null Values

If your base table column accepts null values, you can account for them by one of the following methods:

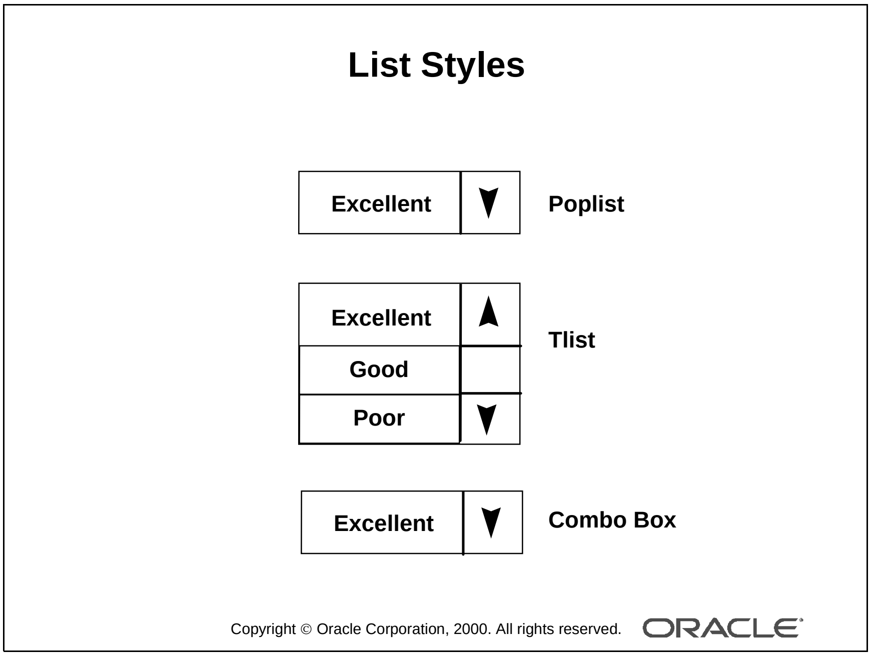
- Set the Check Box Mapping of Other Values property.
- Set the checked or unchecked state to represent null (leave the value blank).
- Ignore a check box in Enter Query mode (use [Shift] + click).



List Items

- Set of mutually exclusive choices, each representing a different value
- Three list styles available
- Space-saving alternative to a radio group
- Smaller-scale alternative to an LOV

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE



List Styles

Excellent ▼ Poplist

Excellent ▲
Good
Poor ▼ Tlist

Excellent ▼ Combo Box

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Creating a List Item

What Is a List Item?

A *list item* is an interface object that displays a predefined set of choices, each corresponding to a specific data value. You use the list item at run time to select a single value. List choices or elements are mutually exclusive; one and only one can be selected at a time.

The Three List Item Styles

List Style	**Description**
Poplist	Appears as a field with an iconic button attached to the right side (When you click a poplist, all its list elements are displayed.)
Tlist	Appears as a rectangular box that displays the list elements (When the display area is not big enough to display all the list elements, a scroll bar is automatically attached to the right side to view the remaining list elements.)
Combo box	Appears as a field with a down arrow next to its right side (Use the button to display all the combo box list elements. The combo box accepts user input.)

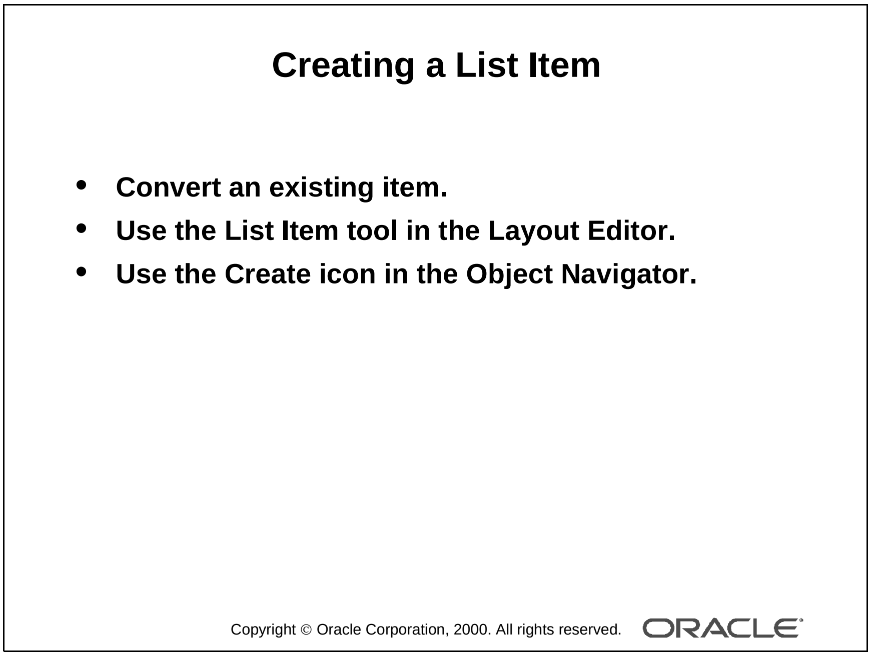
Uses and Benefits of List Items

- Enable display of a defined set of choices
- Display a set of choices without using a vast area of canvas
- Provide an alternative to radio groups
- Provide a Windows-style list of values

Setting the Value for a List Item

The value for a list item can be set in any of the following ways:

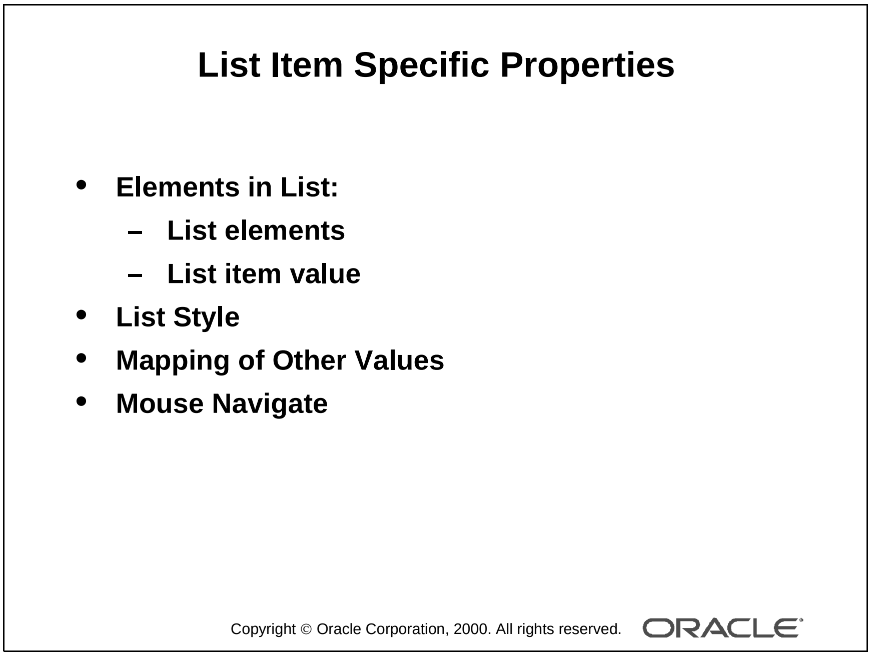
- User selection
- User input (combo box style only)
- A default value
- Programmatic control



Creating a List Item

- Convert an existing item.
- Use the List Item tool in the Layout Editor.
- Use the Create icon in the Object Navigator.

Copyright © Oracle Corporation, 2000. All rights reserved.



List Item Specific Properties

- Elements in List:
  - List elements
  - List item value
- List Style
- Mapping of Other Values
- Mouse Navigate

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a List Item

A list item can be created in three ways:

- Converting an existing item
- Using the List Item tool in the Layout Editor
- Using the Create icon in the Object Navigator

Item Properties Specific to the List Item

Property	**Function**
Elements in List	Opens List Item Elements dialog window (covered later in this lesson)
List Style	Specifies the display style of the list item (choose from Poplist, Tlist, or Combo Box)
Mapping of Other Values	Determines how other values are processed
Mouse Navigate	Determines whether Form Builder navigates to the item and moves input focus to it when the user activates the item with the mouse

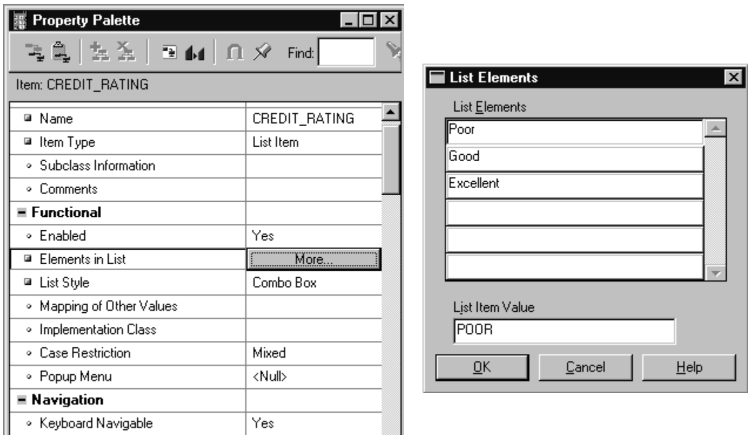
Note: The poplist and combo box take up less space, but end users must open them to see the list elements. A Tlist remains “open,” and end users can see multiple values at a time. Use the attached scroll bar to see more values if the Tlist is not big enough to display all the list elements.

The Elements in List Property

When you click the More property control button for the Elements in List property, the List Item Elements dialog window opens.

Element	**Description**
List Elements	Enables you enter the list elements as they appear at run time
List Item Value	Enables you to specify the actual value that correspond to each of the list elements

Creating a List Item



Item: CREDIT_RATING

Name	CREDIT_RATING
Item Type	List Item
Subclass Information	
Comments	
Functional	
Enabled	Yes
Elements in List	More...
List Style	Combo Box
Mapping of Other Values	
Implementation Class	
Case Restriction	Mixed
Popup Menu	<Null>
Navigation	
Keyboard Navigable	Yes

List Elements

- Poor
- Good
- Excellent

List Item Value: POOR

Copyright © Oracle Corporation, 2000. All rights reserved.

Technical Note

To obtain a list of available functions when defining list elements, select [Ctrl] + k while the input focus is in the List Elements window.

Instructor Note

What are some of the differences between a list item and an LOV?
List items:

- Are generally used for a small number of elements
- Do not have a Find button
- Cannot be attached to other items
- Are not based on a SELECT statement

How to Convert an Existing Item into a List Item

You can convert an existing item into a list item by changing its Item Type property to List Item and setting the relevant properties.

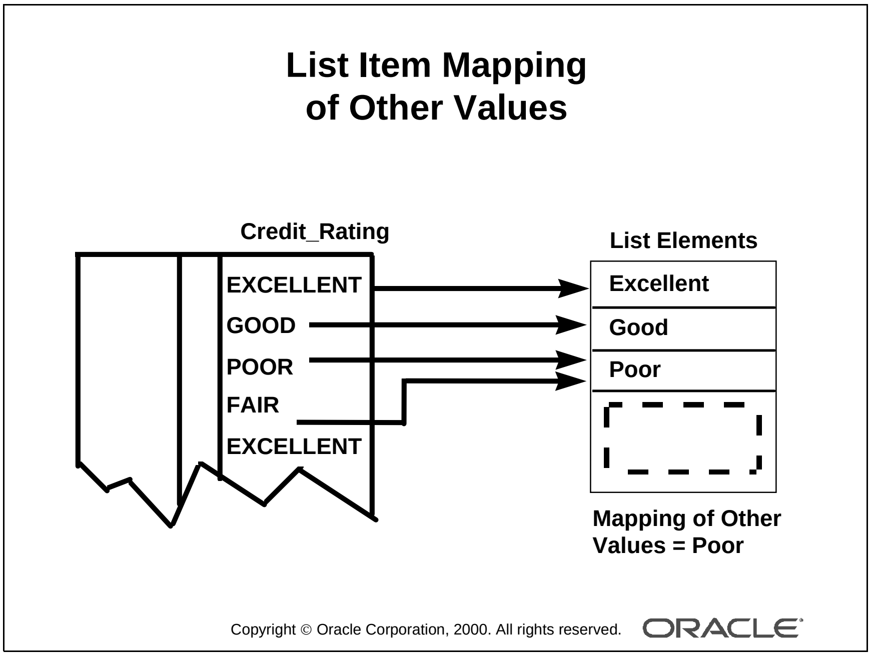
1. Invoke the Property Palette for the item that you want to convert.
2. Set the Item Type property to List Item.
3. Select the Elements in List property.
4. Click More.

   The List Item Elements dialog box appears.
5. Enter the element that you want to appear in your list item in the List Elements column.
6. Enter the value for the currently selected list element in the List Item Value field.
7. Create additional list elements and values by repeating steps 5 and 6.
8. Click OK to accept and close the List Item Elements dialog box.
9. Set the Other Values property to do one of the following:
   - Reject values other than those predefined as list values
   - Accept and default all other values to one of the predefined list element values
10. Enter an initial value for the list item.

How to Create a List Item in the Layout Editor

You can also create a list item by using the List Item tool in the Layout Editor.

1. Invoke the Layout Editor.
2. Set the canvas and block to those on which you want the list item to be displayed.
3. Select the List Item tool.
4. Click the canvas in the position where you want the list item to be displayed.
5. Double-click the list item to invoke its Property Palette.
6. Set the properties as required.



List Item Mapping of Other Values

Credit_Rating

EXCELLENT

GOOD

POOR

FAIR

EXCELLENT

List Elements

Excellent

Good

Poor

Mapping of Other Values = Poor

Copyright © Oracle Corporation, 2000. All rights reserved.

NULL Values in a List Item

If the base table column for a list item accepts NULL values, Form Builder creates a pseudochoice in the list to represent the null.

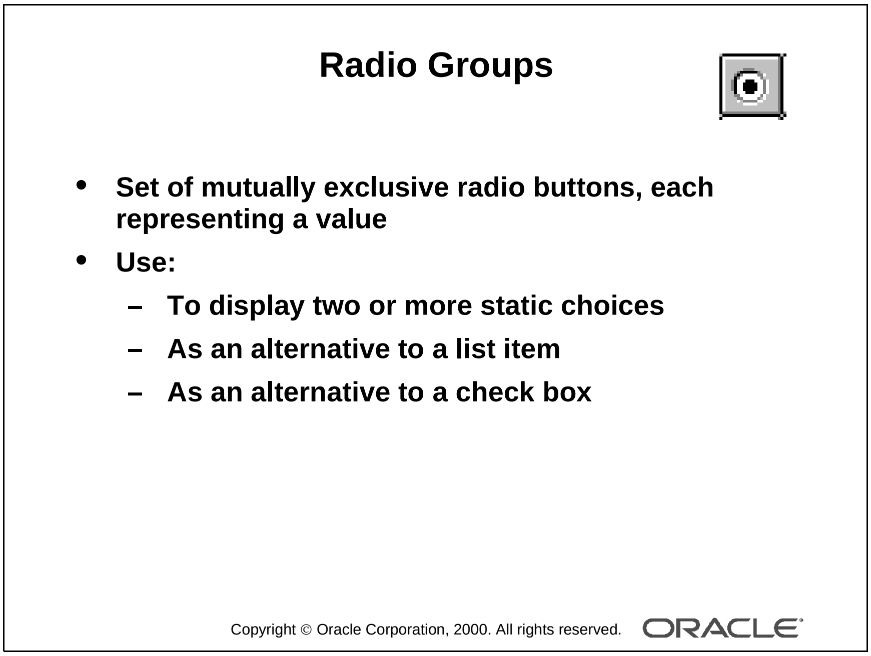
All three list styles display a blank field if a query returns a NULL value. If the Data Required property is set to Yes, upon activation list items display a blank element.

- A poplist displays a blank element for a NULL value.
- For Tlists, the user must scroll through to display the blank element.
- A combo box does not display a blank element. The end user must delete the default value if the default value is not NULL.

Handling Other Values in a List Item

If the base table column for a list item accepts values other than those associated with your list elements, you must specify how you want to handle the values. Do this in one of the following ways:

- Ignore other values by leaving the Mapping of Other Values property blank.
- Associate the other values with one of the existing list elements (by naming either the list element or its associated value) in the Mapping of Other Values property.



Radio Groups

- Set of mutually exclusive radio buttons, each representing a value
- Use:
  - To display two or more static choices
  - As an alternative to a list item
  - As an alternative to a check box

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Creating a Radio Group

What Is a Radio Group?

A *radio group* is a set of radio buttons. Each radio button represents a different value. These values and hence their corresponding radio buttons are mutually exclusive.

Uses and Benefits of Radio Groups

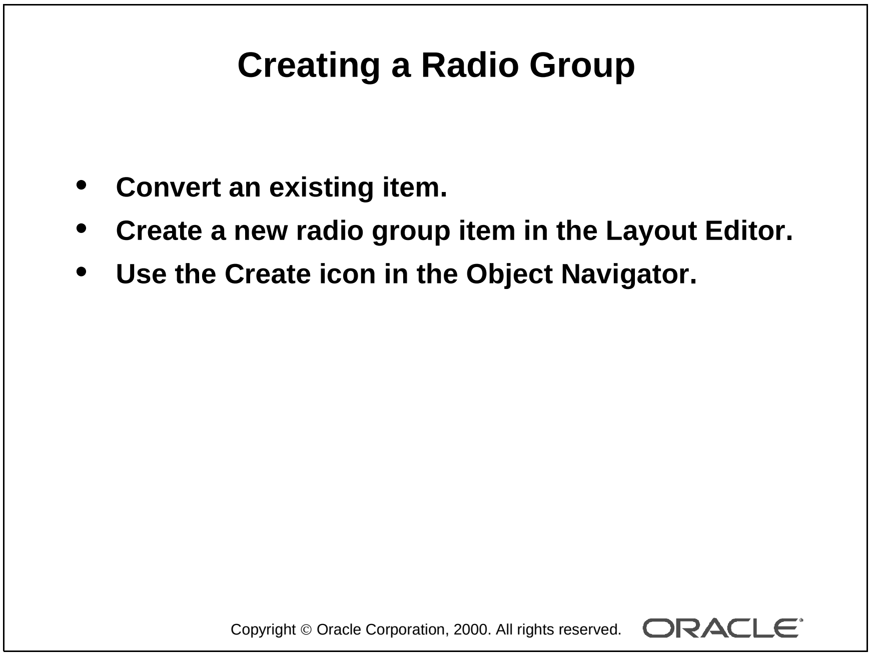
- Provide a choice between two or more static values
- Provide an alternative to list items with two or three choices
- Provide a choice between two alternatives, where choice is not On/Off or Yes/No; for example, Landscape or Portrait print format

Note: Consider list items instead of radio groups if there are more than four or five choices.

Using a Radio Group at Run Time

You can do the following at run time:

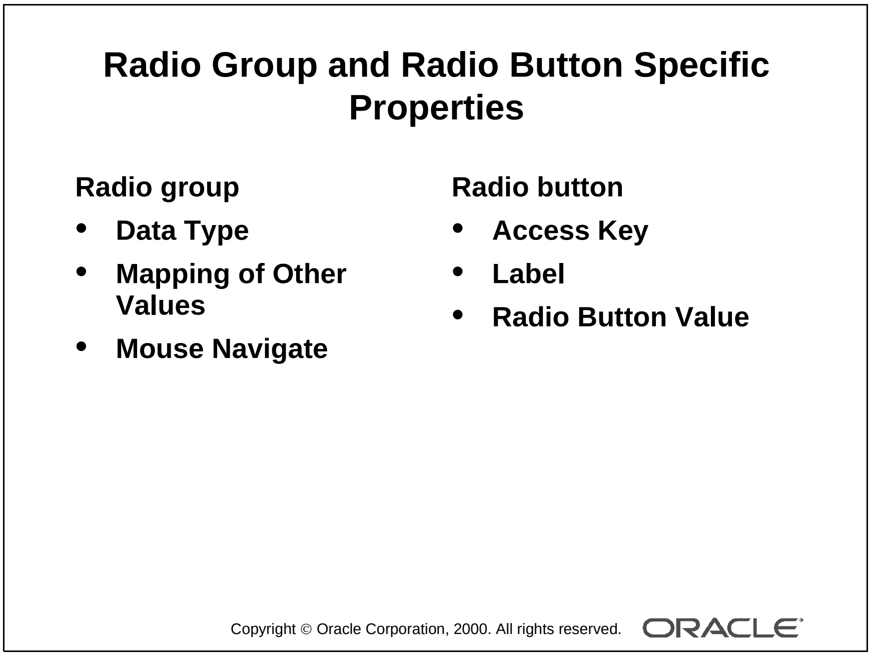
- Set radio group values:
  - By user input
  - By means of the Initial Value property
  - Programmatically
- Query individual radio button values
- Ignore radio button values in Enter Query mode by ensuring that none are selected



Creating a Radio Group

- Convert an existing item.
- Create a new radio group item in the Layout Editor.
- Use the Create icon in the Object Navigator.

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE



Radio Group and Radio Button Specific Properties

Radio group

- Data Type
- Mapping of Other Values
- Mouse Navigate

Radio button

- Access Key
- Label
- Radio Button Value

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Creating a Radio Group

A radio group can be created in three ways:

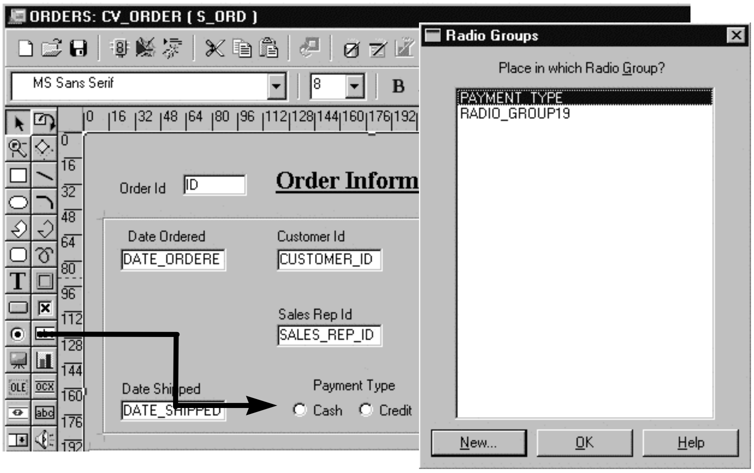
- Converting an existing item to a radio group
- Creating a new radio group item in the Layout Editor
- Using the Create icon in the Object Navigator

Item Properties Specific to Radio Group Items and Radio Buttons

Radio Group Property	**Function**
Data Type	Sets the data type (There is a restricted choice of CHAR, NUMBER, and DATE.)
Mapping of Other Values	Determines how values other than those specified are processed
Mouse Navigate	Determines whether Form Builder navigates to the item when the operator activates the item with the mouse

Radio Button Property	**Function**
Name	Identifies each radio button in the radio group
Access Key	Determines which combination of keys can be used to navigate to and manipulate this button
Label	Specifies the text that appears adjacent to the radio button (These labels are independent of the button values.)
Radio Button Value	Specifies the value that the radio button represents in a radio group

Creating Radio Buttons



Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The canvas property for a radio group is set in the Property Palette of the radio group. The individual radio buttons do not have a canvas property.

How to Convert an Existing Item into a Radio Group

You can convert an existing item to a radio group by changing the item type and setting the properties for a radio group.

1. Invoke the Property Palette for the item that you want to convert.
2. Set the Item Type property to Radio Group.
3. Set the Mapping of Other Values property to specify how the Radio Group should handle any other values.
4. Set the Initial Value property, as required. This should be the name of a radio button.
5. Expand the item node in the Object Navigator.

   The Radio Buttons node appears.
6. Select the Radio Buttons node and click the Create icon.

   A radio button displays in the Object Navigator and the Property Palette takes on its context.
7. Enter a name, value, and a label for the radio button.
8. Specify the display properties of the radio button.
9. Create additional radio buttons by repeating steps 6 through 8.

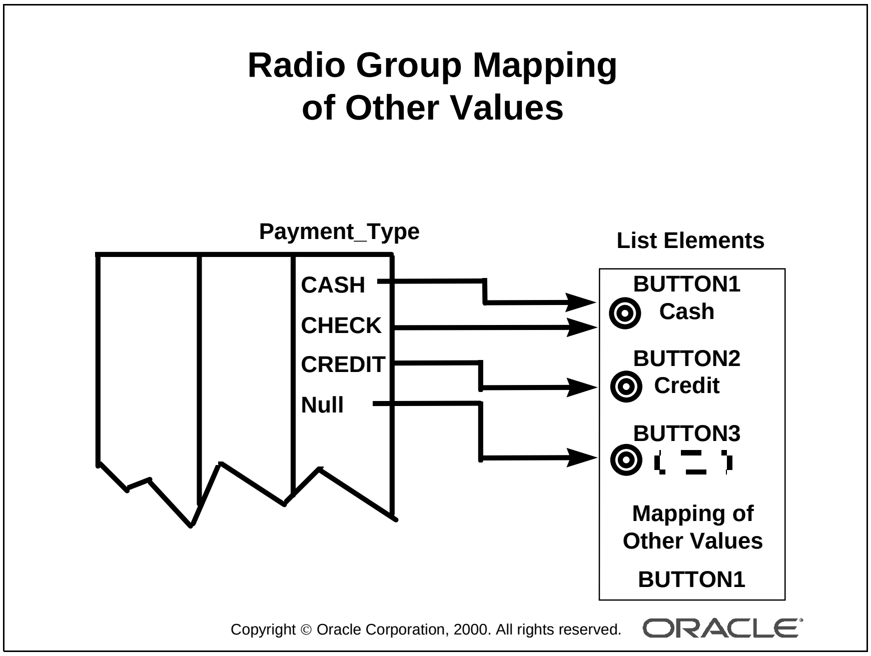
How to Create a Radio Group in the Layout Editor

You can also create a radio group by using the Radio Button tool in the Layout Editor.

1. Invoke the Layout Editor.
2. Set the canvas and block to those on which you want the radio group to be displayed.
3. Select the Radio Button tool.
4. Position the cursor at the desired location and click.

   If you already have a radio group in the current block, the Radio Groups dialog box appears and you must decide whether the new radio button should appear in the existing group or a new one.
5. Double-click the radio button to invoke the Property Palette.

   The new radio group is created implicitly.
6. Set the radio button properties as required.



Radio Group Mapping of Other Values

Payment_Type

CASH

CHECK

CREDIT

Null

List Elements

BUTTON1
Cash

BUTTON2
Credit

BUTTON3

Mapping of
Other Values

BUTTON1

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

You must specify a valid initial value, except under either of the following conditions:

- The radio group accepts other values.
- The value associated with one of the radio buttons is NULL.

NULL Values in a Radio Group

A radio group can treat NULL as a valid value. You should account for the NULL case, if your base table column allows them. Do this in one of the following ways:

- Use the Mapping of Other Values property to implicitly force NULL to a radio button.
- Assign the NULL to its own radio button.

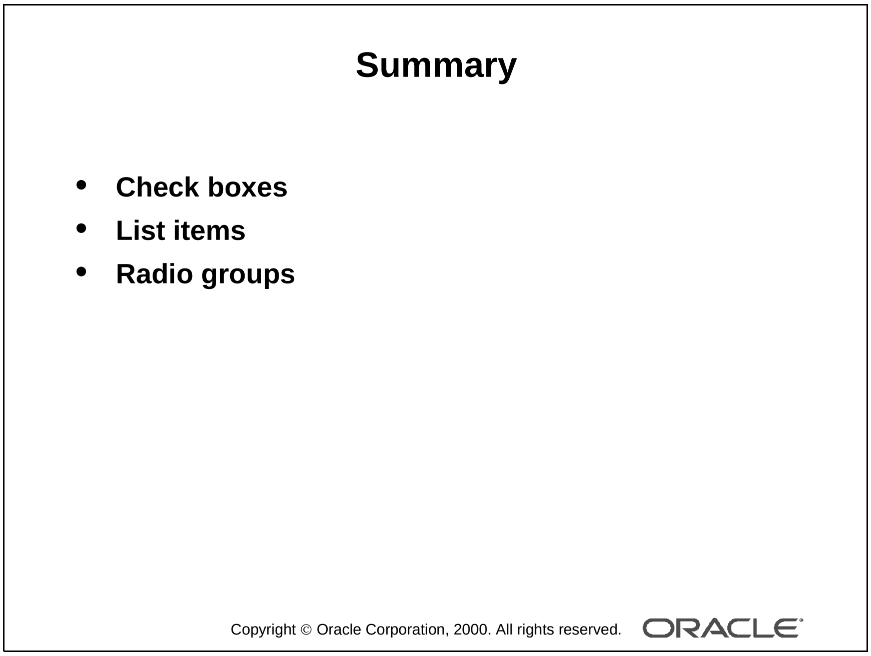
  Note: To assign a NULL value to a radio button, leave the Radio Button Value property blank.

Handling Other Values in a Radio Group

If the base table column for a radio group accepts values other than those associated with your radio buttons, you must use one of the following methods to specify how you want to handle the values:

- Ignore other values (by leaving the radio group's Mapping of Other Values property blank)
- Associate the other values with one of the existing radio buttons (by naming the associated value of the button in the Mapping of Other Values property)

Note: Ignoring other values results in the entire row being ignored during query processing.



Summary

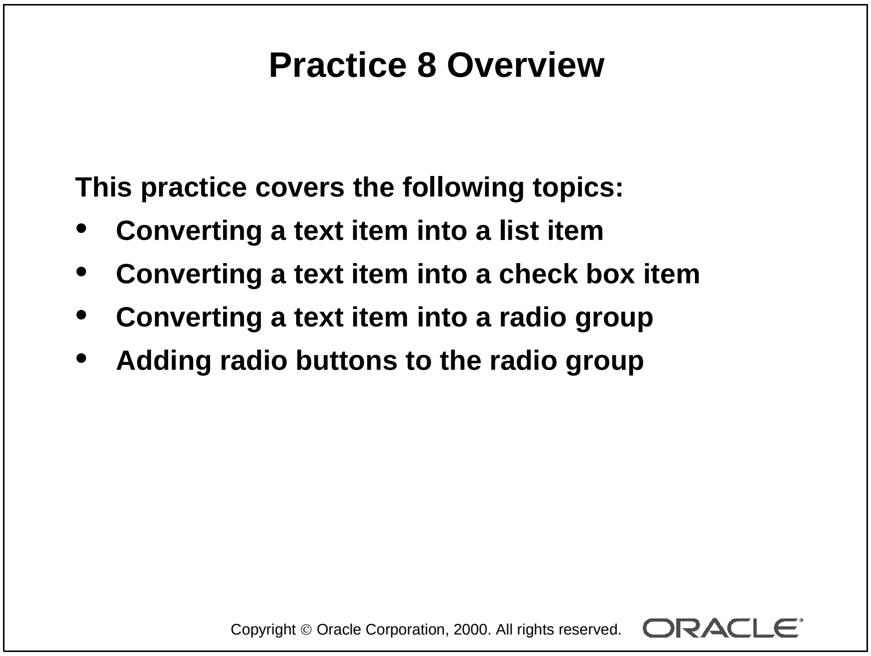
- **Check boxes**
- **List items**
- **Radio groups**

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

In this lesson, you learned how to create items that accept direct user input. Use these items to enhance the user interface:

- Check boxes: To convert items that have two possible states
- List items (Poplists, Tlists, and Combo boxes): To convert items that are mutually exclusive
- Radio groups: To convert items (two or three alternatives) that are mutually exclusive



Practice 8 Overview

This practice covers the following topics:

- **Converting a text item into a list item**
- **Converting a text item into a check box item**
- **Converting a text item into a radio group**
- **Adding radio buttons to the radio group**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 8 in Appendix A, “Practice Solutions.”

Practice 8 Overview

In this practice session, you will convert existing text items into other input item types. You will create a list item, a check box, and a radio group.

- In the CUSTOMERS form, convert the Credit_Rating item into a list item. Save and run the form.
- In the ORDERS form, convert the Order_Filled item into a check box item.
- In the ORDERS form, convert the Payment_Type item into a radio group. Add two radio buttons in the radio group. Save and run the form.

Practice 8

1 In the CUSTG*XX* form, convert the Credit_Rating text item into a pop-up list item.

Add list elements of Poor, Good, and Excellent to represent database values of POOR, GOOD, and EXCELLENT.

Display any other values as POOR.

Ensure that new records display the initial value GOOD.

Resize the poplist in the Layout Editor, so that the elements do not truncate at run time.

2 Save and compile the form.

Deploy your form on the Web to test the changes.

3 In the ORDG*XX* form, convert the Order_Filled text item into a check box.

Set the checked state to represent the base table value of Y and the unchecked state to represent N.

Ensure that new records are automatically assigned the value N.

Allow only those records with Order_Filled values of Y or N to display.

Remove the existing prompt and set label as Order Filled.

In the Layout Editor, resize the check box so that its label is displayed to the right.

4 Convert the Payment_Type text item into a radio group.

Add radio buttons for Cash and Credit to represent database values of CASH and CREDIT.

Define access keys of S for cash and T for credit.

Add text Payment type to describe the radio group’s purpose.

Set Label to Cash for Cash radio button and Credit for Credit radio button.

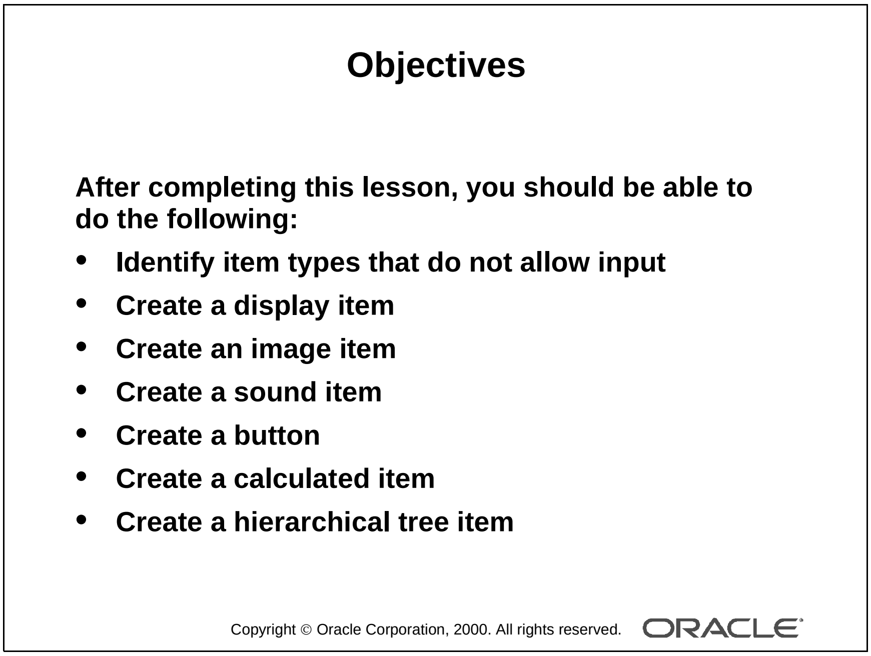
Ensure that new records display the default of Cash.

5 Reorder the items of the S_ORD block in the Object Navigator. Use the order of the items in the Layout Editor as a guide.

6 Save, compile, and run the form to test the changes.

9

Creating Noninput Items



Objectives

After completing this lesson, you should be able to do the following:

- **Identify item types that do not allow input**
- **Create a display item**
- **Create an image item**
- **Create a sound item**
- **Create a button**
- **Create a calculated item**
- **Create a hierarchical tree item**

Copyright © Oracle Corporation, 2000. All rights reserved.

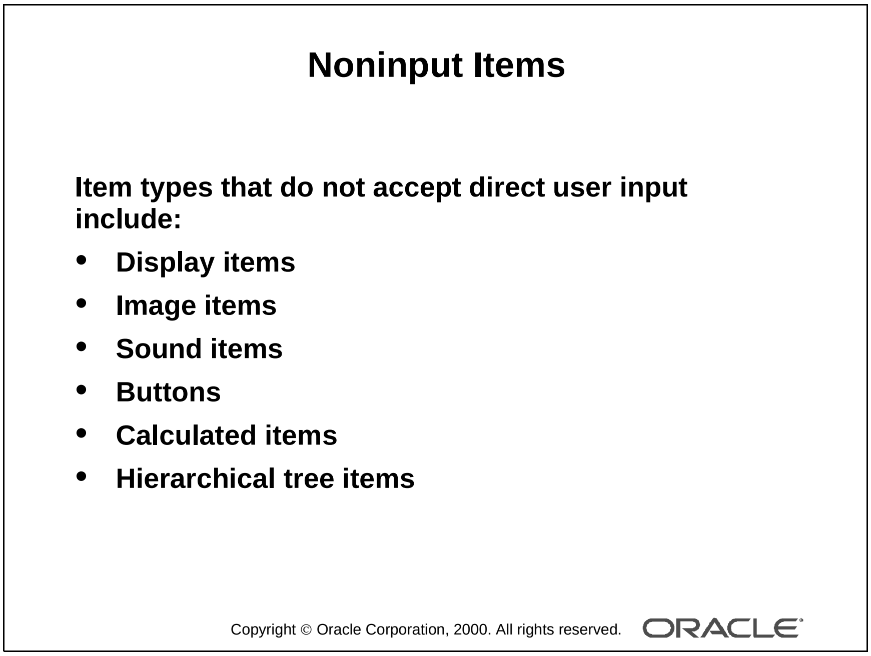
Schedule

Topic	**Timing**
Lecture	60 minutes
Practice	40 minutes
Total	100 minutes

Introduction

Overview

Some Oracle Forms Developer item types do not accept user input (noninput items); however, they do provide an effective means of accessing data and initiating actions. This lesson describes how to create and use noninput items.



Noninput Items

Item types that do not accept direct user input include:

- **Display items**
- **Image items**
- **Sound items**
- **Buttons**
- **Calculated items**
- **Hierarchical tree items**

Copyright © Oracle Corporation, 2000. All rights reserved.

What Are Noninput Items?

Noninput items is a generic term for item types that do not accept direct user input. However, you can set the value of some noninput items by programmatic control. Noninput items can be divided into two groups—those that can display data and those that cannot.

Noninput Items That Can Display Data

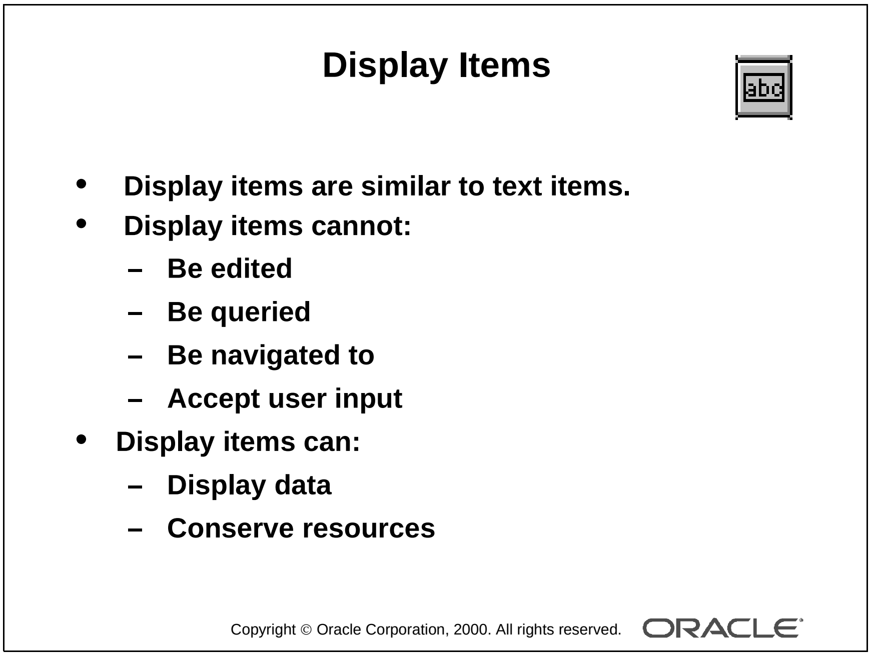
- Display items
- Image items
- Calculated items

Noninput Items That Cannot Display Data

- Sound items
- Push Buttons

Using Noninput Items

Use noninput items to enhance your application by displaying additional data often from a nonbase table.



Display Items

- **Display items are similar to text items.**
- **Display items cannot:**
  - **Be edited**
  - **Be queried**
  - **Be navigated to**
  - **Accept user input**
- **Display items can:**
  - **Display data**
  - **Conserve resources**

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Display Item

What Is a Display Item?

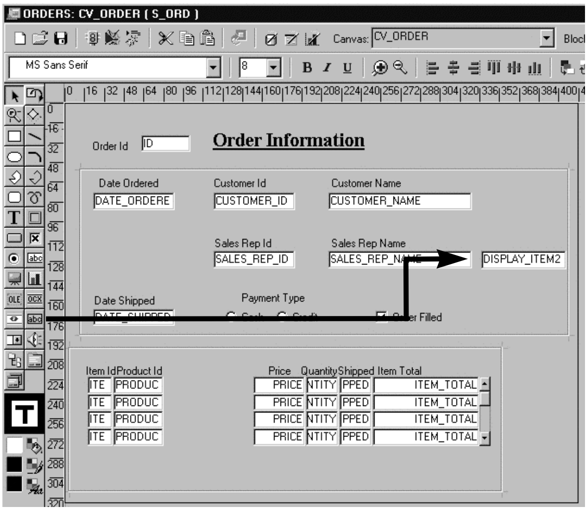
A *display item* is similar to a text item, except that it cannot be edited or navigated to at runtime. A display item is a read-only text box whose value must be fetched or assigned programmatically.

Display items:

- Display additional, no-base table information
- Display derived data values
- Conserve memory

Note: Display items require less memory than text items. However, this is no longer as significant, now that we have the Rendered property for text items.

Creating a Display Item



Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Display Item

A display item can be created by using:

- The Layout Editor
- The Create icon in the Object Navigator
- The Item Type property to convert an existing item into a display item

Whichever method you choose, you need to set the required item properties in the Property Palette.

How to Create a Display Item from the Layout Editor

1. Invoke the Layout Editor.
2. Display the desired canvas and ensure that the correct data block is set.
3. Select the Display Item tool.
4. Click the canvas at the position where the display item is required.
5. Double-click the new display item.

   The Property Palette displays.
6. Change the name from DISPLAY_ITEM*XX* to the required name.
7. Specify the other properties as required.

Note: Remember to set the Database Item property to No for a display item whose value is not stored in the base table.

You can assign a format mask to a single-line display item by manipulating its Format Mask property.

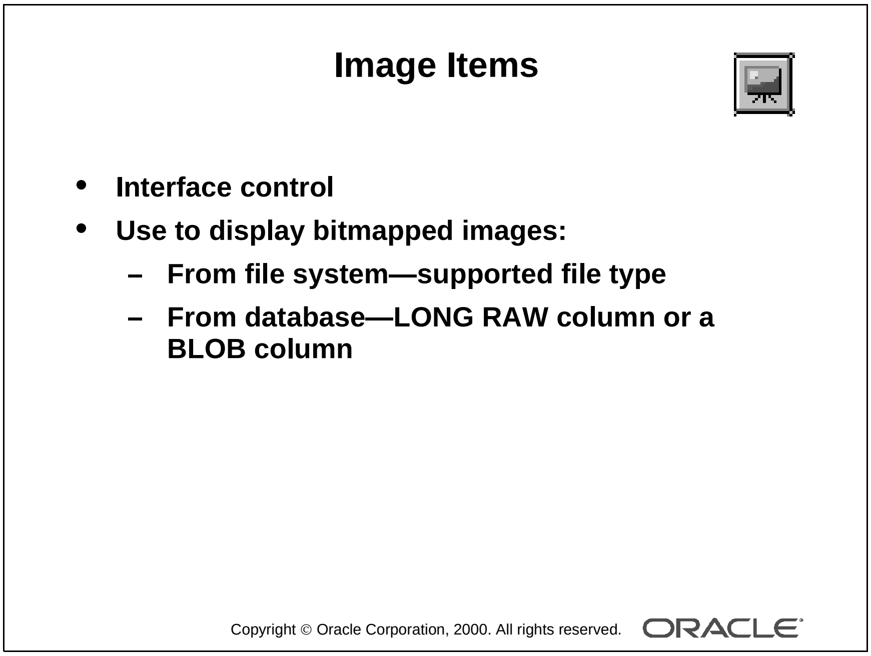


Image Items

- **Interface control**
- **Use to display bitmapped images:**
  - **From file system—supported file type**
  - **From database—LONG RAW column or a BLOB column**

Copyright © Oracle Corporation, 2000. All rights reserved.

Technical Note

You can also populate an image item with a BFILE, but you will need to use DBMS_LOB to do so.

Creating an Image Item

You can use images as graphic objects within a form module. A graphics image displays automatically and cannot be manipulated at run time. It can be imported from the database or the file system.

Alternatively, you can display images within image items.

What Is an Image Item?

An *image item* is a special interface control that can store and display vector or scanned bitmapped images. Just as text items store and display VARCHAR2, number, or date values, image items store and display images.

Like other items that store values, image items can be either data block items or control items.

Displaying Image Items

You can populate an image item by using one of the following methods:

- Fetching from a LONG RAW or BLOB database column
- Using a trigger and a built-in to populate the image item programmatically
- Cutting or pasting an image to the clipboard, selecting the image item at runtime, and choosing Edit—>Paste

Storing Images

You can store images in either the database or the file system.

When you insert images into the database by means of a Form Builder save (commit), they are automatically compressed using Oracle Image compression.

Where Image Is Stored	Description
Database	Long Raw column compressed image that can be up to four gigabytes
File	Any supported file format

Note: To conserve client memory when displaying large image items, reduce the number of records that are buffered by manipulating the Number of Records Buffered data block property.

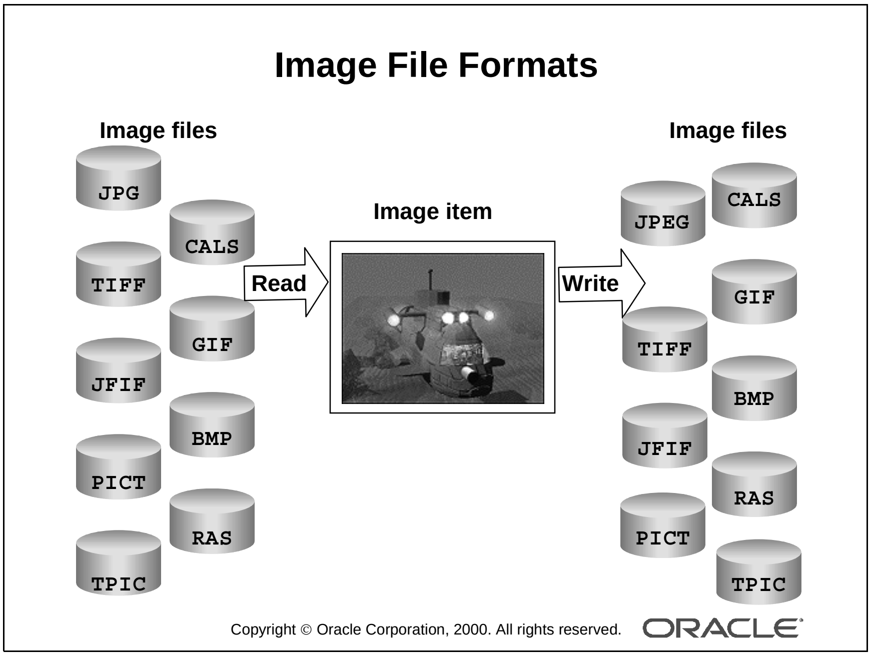


Image File Formats

Copyright © Oracle Corporation, 2000. All rights reserved.

Image File Formats

Form Builder supports the following image formats:

File Suffix	Description	Image Items
BMP	Microsoft Windows and OS/2 BitMap Picture	Read/Write
CALS	CALS type raster	Read/Write
GIF	CompuServe	Read/Write
JFIF	JPEG File Interchange Format	Read/Write
TIFF	Tag Image File Format	Read/Write
JPEG	Joint Photographic Experts Group	Read/Write
PICT	Macintosh Quickdraw Picture	Read/Write
RAS	Sun Raster	Read/Write
TPIC	Truevision Raster Graphics Array Picture	Read /Write

Web Design Tip

To reduce network traffic, limit the number of image items and background images in your Internet applications. Each time an image is required, it must be downloaded from the application server; images cannot be cached.

For example, to display a company logo in your application, you could include the image in the HTML page that downloads at application startup rather than retrieving the image from the database or the file system. The HTML page can be cached.

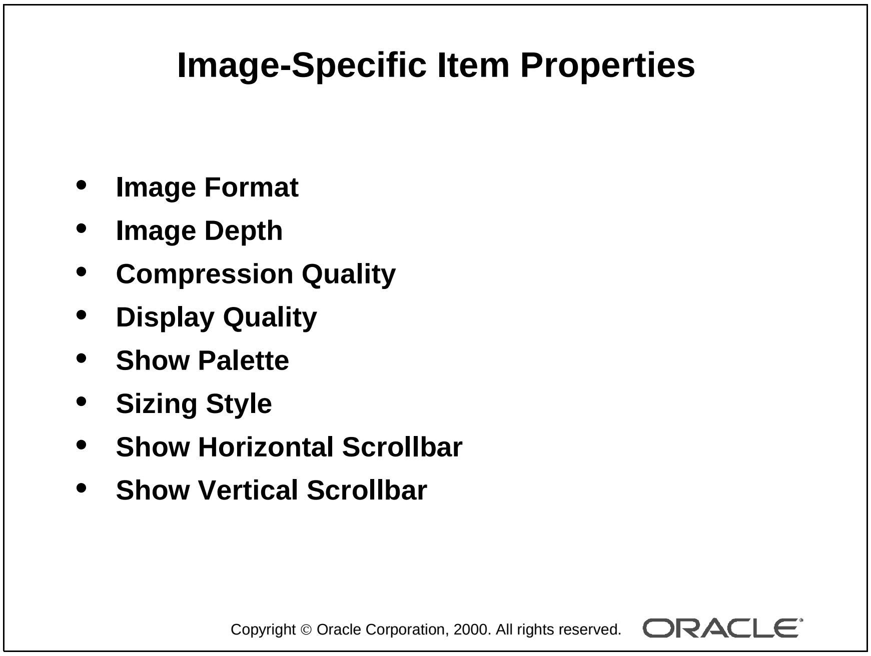


Image-Specific Item Properties

- Image Format
- Image Depth
- Compression Quality
- Display Quality
- Show Palette
- Sizing Style
- Show Horizontal Scrollbar
- Show Vertical Scrollbar

Copyright © Oracle Corporation, 2000. All rights reserved.

Item Properties Specific to the Image Item

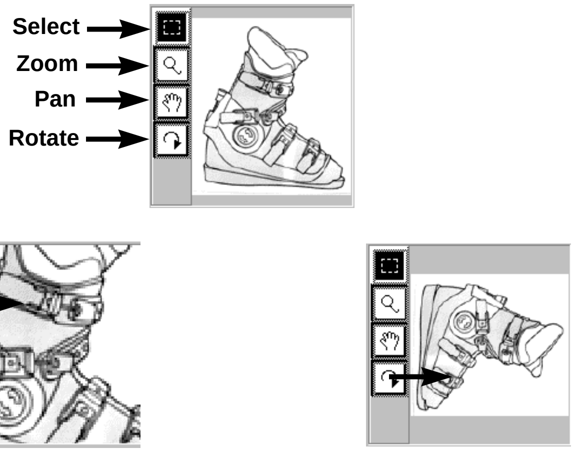
Property	**Function**
Image Format	Specifies the format in which the image item will be stored in the database
Image Depth	Specifies the image depth setting for the image item being read from or written to a file in the file system (Choose from: Original, Monochrome, Gray, LUT, or RGB.)
Compression Quality	Specifies whether an image item being read from or written to a file should be compressed, and to what degree (Choose from: None, Minimum, Low, Medium, High, or Maximum.)
Display Quality	Determines the resolution used to display the image item (You can use this property to control the trade off between quality and performance. Choose from: High, Medium, or Low.)
Show Palette	Display image manipulation palette (Discussed later in this lesson).
Sizing Style	Determines how much of the image displays when the image size does not match the size of the item (Crop and Adjust are the two style choices. Crop cuts the edges off the images so that it fits in the rectangle. Adjust scales the image to fit within the display rectangle.)
Show Horizontal Scrollbar	Displays a horizontal scroll bar
Show Vertical Scrollbar	Displays a vertical scroll bar

Note: Image items do not have a Data Type property. If you set an image item Database Item property to Yes, Form Builder understands that the data type is LONG RAW.

Instructor Note

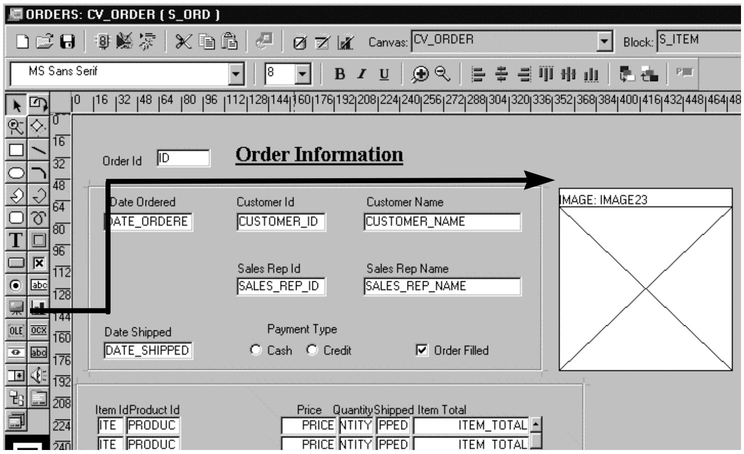
- LUT = Lookup Table
- RGB = Red, Green, Blue

Image Manipulation Palette



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE®

Creating an Image Item



ORDERS: CV_ORDER (S_ORD)

Canvas: CV_ORDER Block: S_ITEM

MS Sans Serif 8

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE®

Manipulating an Image

To manipulate an image at run time, set the Show Palette property for the image item to Yes. This will display a palette adjacent to the image item with three image manipulation tools.

Tool	**Function**
Select	Select an area in the image
Zoom	Zoom in or zoom out the image incrementally
Pan	Pan unseen portions of the image
Rotate	Rotate the image clockwise in 90-degree increments

Note: To zoom out, hold down the [Shift] key when you select the zoom tool.

Creating an Image Item

An image item can be created in three ways:

- By using the Image Item tool in the Layout Editor (as described in the next section)
- By using the Create icon in the Object Navigator
- By converting an existing item into an image item

Steps to Create an Image Item from the Layout Editor

1. Invoke the Layout Editor.
2. Set the canvas and block to those on which you require the item to display.
3. Select the Image Item tool.
4. Click the canvas at the position where you want the image item to display.
5. Double-click the image item.

   The Property Palette displays.
6. Change the name from IMAGE*XX* to the required name.
7. Specify the other properties as required.

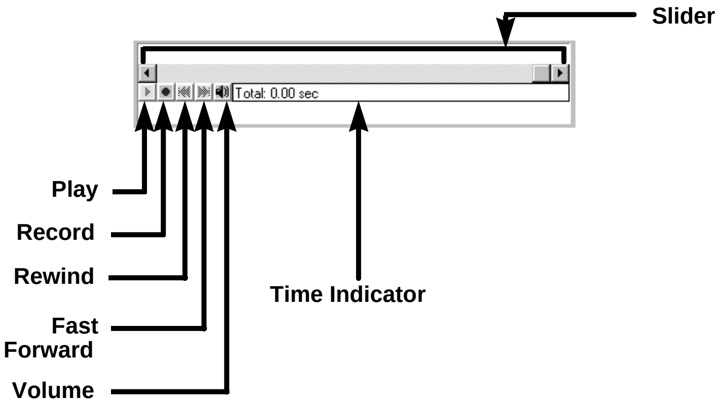
Note: Remember to set the Database Item property to No for an image item whose value is not stored in the base table.

Sound Items

- **Interface control**
- **Use to play and record sound:**
  - **From file system—supported sound types**
  - **From database—LONG RAW column or BLOB column**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE®

Sound Item Control



Slider

Play

Record

Rewind

Fast Forward

Volume

Time Indicator

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE®

Creating a Sound Item

What Is a Sound Item?

A *sound item* is a special interface control that can play and record sound data. Sound data can be stored in either the database or a file system.

Sound items can be either data block items or control items.

Playing Sound Data

You can play sound data in one of two ways:

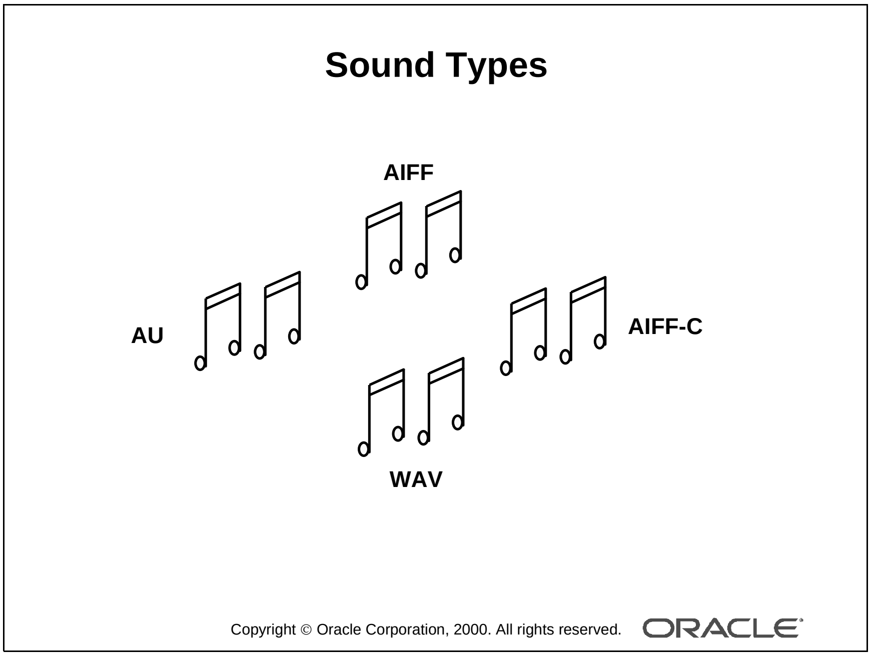
- Fetching from a LONG RAW or BLOB database column
- Programmatically using a trigger and built-ins

Sound Item Control

When you create a sound item, Form Builder automatically represents the item in the layout with a sound control widget for user interaction. Each component within the sound control widget has a specific task.

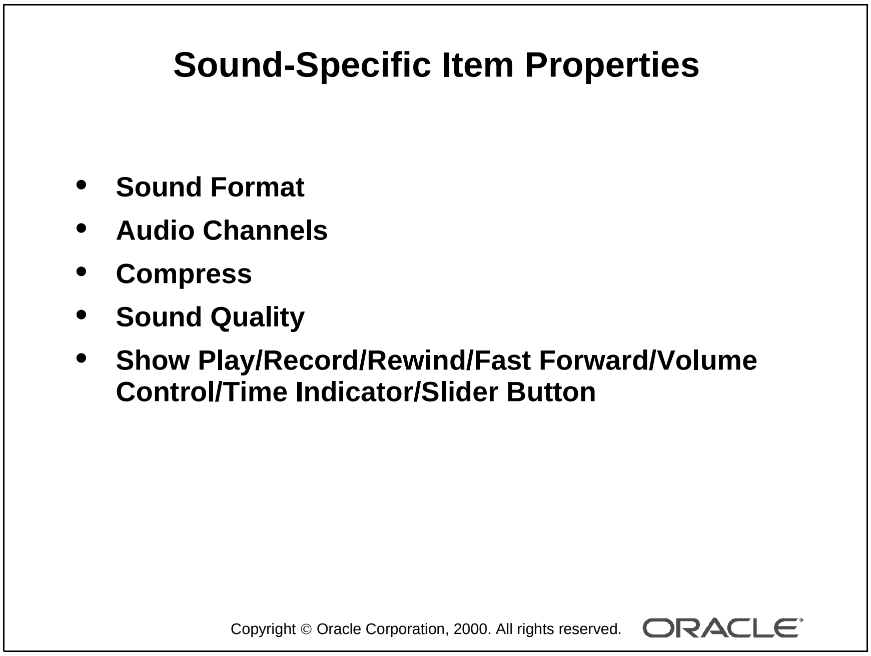
Sound Control	**Description**
Play	Starts playback of sound from the current pointer position
Record	Starts recording sound data from the current pointer position
Rewind	Rewinds sound data
Fast Forward	Fast forwards sound data
Volume	Displays a vertical slide control you can use to control playback and recording volume
Time Indicator	Displays three modes of time information for the sound data (You can click the indicator text to switch between total time, elapsed time, and remaining time.)
Slider	Displays a horizontal slide control to indicate current pointer position within the sound data (You can slide the handle left or right to reposition the pointer.)

Note: Each of the sound control components is optional, except for Play or Record, one of which must always be displayed if the sound item is visible.



Sound Types

Copyright © Oracle Corporation, 2000. All rights reserved.



Sound-Specific Item Properties

- **Sound Format**
- **Audio Channels**
- **Compress**
- **Sound Quality**
- **Show Play/Record/Rewind/Fast Forward/Volume Control/Time Indicator/Slider Button**

Copyright © Oracle Corporation, 2000. All rights reserved.

Sound Types

Form Builder supports the following sound types:

- AU
- AIFF
- AIFF-C
- WAV

Creating a Sound Item

A sound item can be created in three ways:

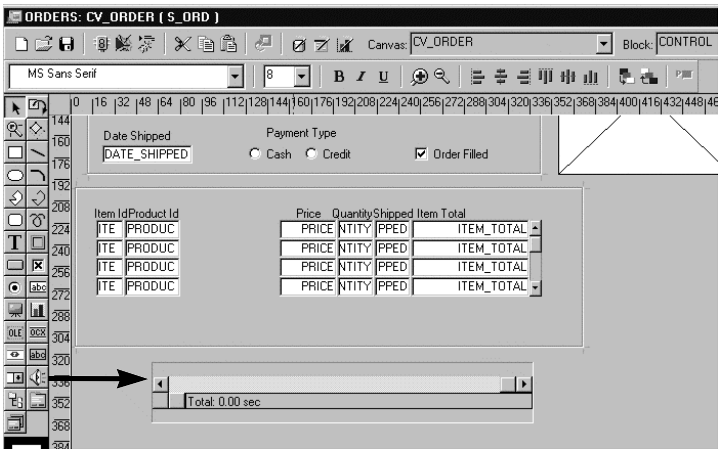
- By converting an existing item
- By using the Sound Item tool in the Layout Editor
- By using the Create icon in the Object Navigator

Item Properties Specific to the Sound Item

Property	Function
Sound Format	Specifies the format in which the sound item will be stored in the database (Choose from AU, AIFF, AIFF-C, or WAVE.)
Audio Channels	Specifies the number of channels with which the sound item will be stored in the database (Choose from Automatic, Mono, or Stereo.)
Compress	Specifies whether a sound object that is read from a file should be compressed
Sound Quality	Specifies the quality for storing the sound item in the database (Choose from Automatic, Highest, High, Medium, Low, or Lowest.)
Show Play/Record/Rewind/Fast Forward/Volume Control/Time Indicator/Slider Button	To display or hide a sound item control component

Note: Large sound objects can degrade performance over a network. Improve performance by setting the Update Changed Columns Only property to Yes for the sound item block.

Creating a Sound Item



ORDERS: CV_ORDER (S_ORD)

Canvas: CV_ORDER

Block: CONTROL

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The built-ins that are relevant to sound items are covered in a later lesson.

How to Create a Sound Item from the Layout Editor

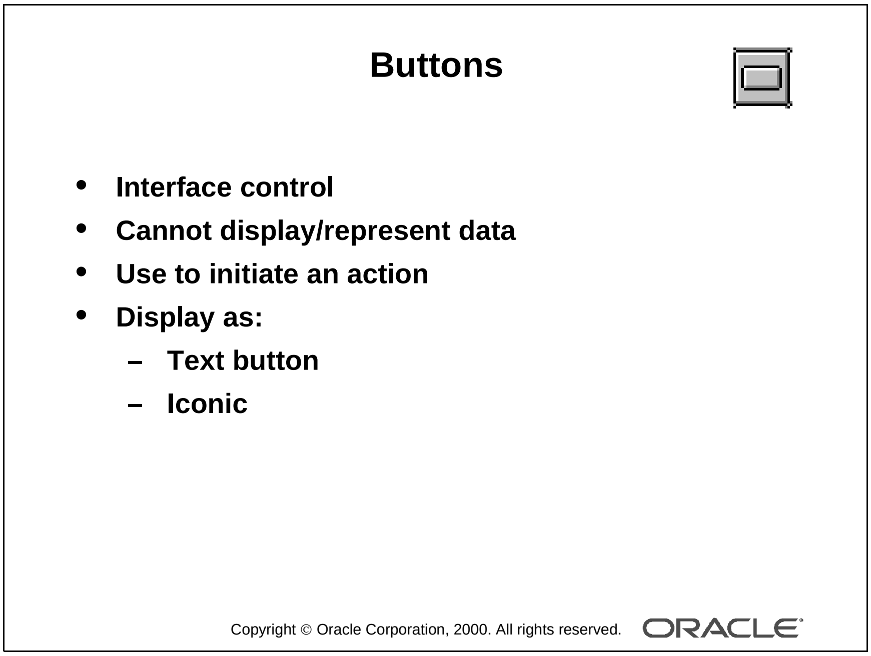
1. Invoke the Layout Editor.
2. Set the canvas and block to those you require the item to display on.
3. Select the Sound Item tool.
4. Click the canvas at the position in which you want the sound item to display.
5. Double-click the sound item.

   The Property Palette displays.
6. Change the name from SOUND_ITEM*XX* to the required name.
7. Specify the other properties as required.

Note: Remember to set the Database Item property to No for a sound item whose value is not stored in the base table.

Web Design Tip

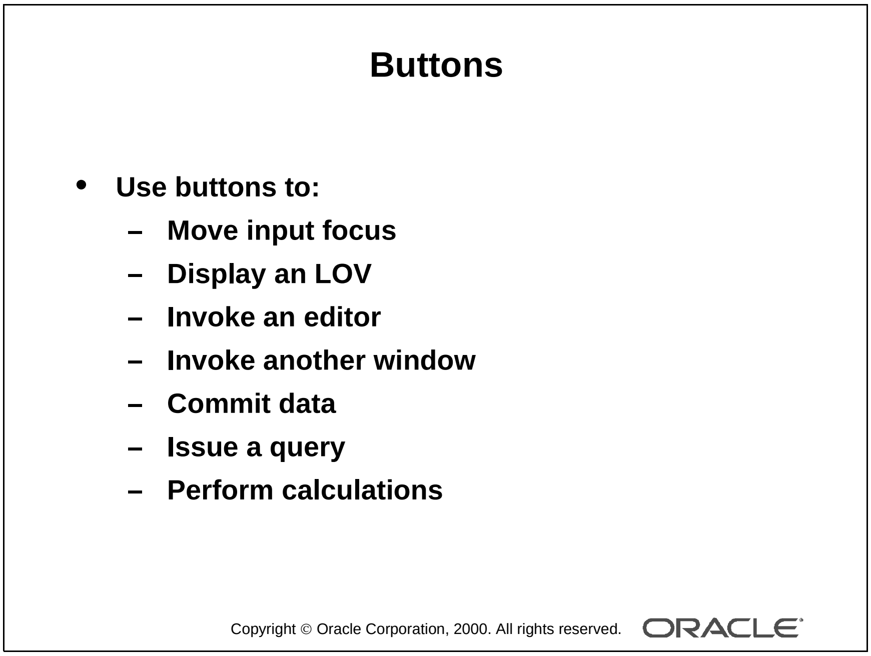
Sound items are not supported in Forms deployed on the Web.



Buttons

- **Interface control**
- **Cannot display/represent data**
- **Use to initiate an action**
- **Display as:**
  - **Text button**
  - **Iconic**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE



Buttons

- **Use buttons to:**
  - **Move input focus**
  - **Display an LOV**
  - **Invoke an editor**
  - **Invoke another window**
  - **Commit data**
  - **Issue a query**
  - **Perform calculations**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Creating a Push Button

What Is a Push Button?

A *push button* is an interface object that you click to initiate an action. A push button is usually displayed as a rectangle with a descriptive label inside. Push buttons cannot store or display values.

You can enhance your form module further by adding push buttons to provide quick and direct access to the most needed operations.

Push Button Styles

Form Builder supports two push button styles:

- Text button: Displayed with a text label on the push button
- Iconic button: Displayed with a bitmapped graphic on the push button, and often used in toolbars

Some Typical Push Button Actions

- Moving the input focus
- Displaying an LOV
- Invoking an editor
- Invoking another window
- Committing data
- Issuing a query
- Performing calculations

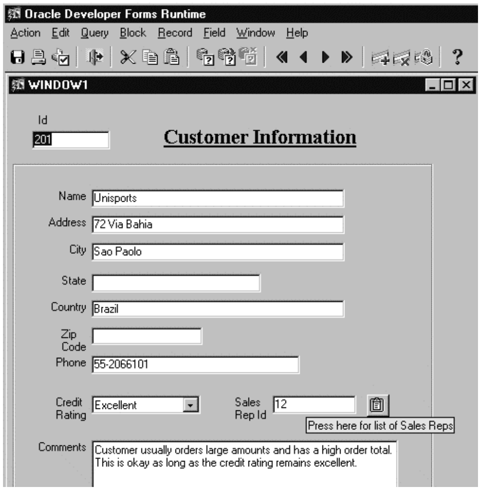
Note: Push buttons do not accept input focus on some window managers. On these platforms, the Keyboard Navigable property has no effect, and users can only interact with the items by using a mouse. Clicking a push button does not move the input focus on these platforms. The input focus remains in the item that was active before the push button.

Button-Specific Item Properties

- **Label**
- **Mouse Navigate**
- **Default Button**
- **Iconic**
- **Icon Filename**
- **Tooltip**
- **Tooltip Visual Attribute Group**

Copyright © Oracle Corporation, 2000. All rights reserved.

Tool Tip



Customer Information

Id: 201

Name: Unisports

Address: 72 Via Bahia

City: Sao Paolo

State:

Country: Brazil

Zip Code:

Phone: 55-2066101

Credit Rating: Excellent

Sales Rep Id: 12

Press here for list of Sales Reps

Comments: Customer usually orders large amounts and has a high order total. This is okay as long as the credit rating remains excellent.

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Push Button

A push button can be created by using:

- The Push Button tool in the Layout Editor
- The Create icon in the Object Navigator

Item Properties Specific to the Push Button

Property	**Function**
Label	Specifies the text label that appears on the push button at run time
Mouse Navigate	Determines whether Form Builder navigates to the item when you click on it by using the mouse
Default Button	Determines whether this is the default push button for the block (You can select the default push button implicitly by pressing a platform-specific key without the need to navigate or use the mouse.)
Iconic	Determines whether the push button displays as an icon instead of as a label
Icon Filename	Identifies the name of the file that contains the icon resource (Do *not* enter the icon file extension here.)
Tooltip	Specifies the help text that should appear in a tool tip beneath the current item
Tooltip Visual Attribute Group	Specifies the named visual attribute that should be applied to the tool tip at run time

Note: On some window managers, the default push button is bordered or highlighted in a unique fashion to distinguish it from other push buttons.

Tool Tip

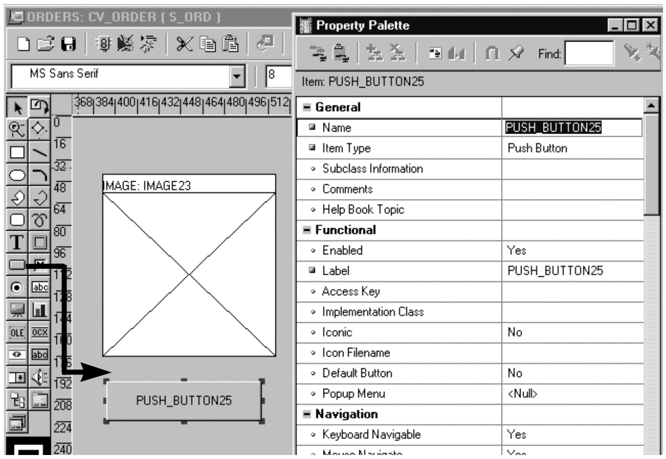
A tool tip is a small text box that displays help information about an item when you navigate to it with the mouse. The tool tip remains visible until the mouse is moved outside the item, or until another window is shown, or until the user performs an action in the current item.

You can specify helpful information about a push button by entering the required help text in the Tooltip property of the push button. At run time, this text will automatically display in a small box when the user navigates to the push button.

Instructor Note

A tool tip is completely portable, and it replaces `HINT.PLL/DLL`. You cannot control the timing of a tool tip as you can with `HINT.PLL` by using timers. It is internalized in the form code.

Creating a Button



Item: PUSH_BUTTON25

General	
Name	PUSH_BUTTON25
Item Type	Push Button
Subclass Information	
Comments	
Help Book Topic	
Functional	
Enabled	Yes
Label	PUSH_BUTTON25
Access Key	
Implementation Class	
Iconic	No
Icon Filename	
Default Button	No
Popup Menu	<Null>
Navigation	
Keyboard Navigable	Yes
Mouse Navigate	Yes

Copyright © Oracle Corporation, 2000. All rights reserved.

How to Create a Push Button from the Layout Editor

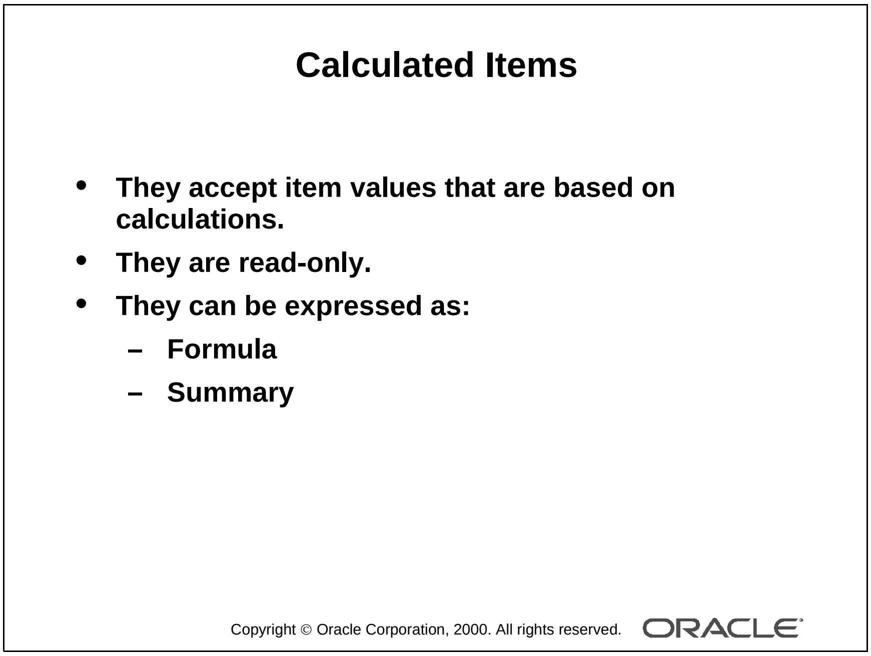
1. Invoke the Layout Editor.
2. Set the canvas and block to those on which you require the push button to display.
3. Select the Push Button tool.
4. Click the canvas at the position where you want the push button to display.
5. Double-click the push button.

   The Property Palette displays.
6. Change the name from PUSH_BUTTON*XX* to the required name.
7. Specify the other properties as required.

Note: You can use the mouse to resize and move the push button once you have created it.

Web Design Tip

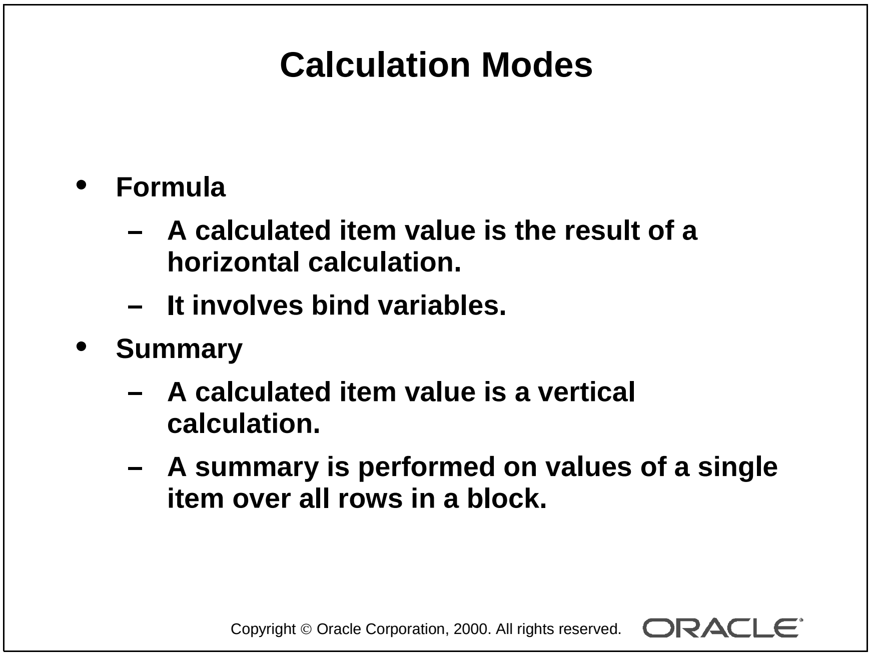
Icon image files for iconic buttons must be in GIF or JPEG format (and not in ICO format) if your form application is going to be deployed on the Web.



Calculated Items

- They accept item values that are based on calculations.
- They are read-only.
- They can be expressed as:
  - Formula
  - Summary

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE



Calculation Modes

- Formula
  - A calculated item value is the result of a horizontal calculation.
  - It involves bind variables.
- Summary
  - A calculated item value is a vertical calculation.
  - A summary is performed on values of a single item over all rows in a block.

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Creating a Calculated Item

What Is a Calculated Item?

With a calculated item you can declaratively base item values on calculations involving one or more variable values. For example, obtaining a running total of employees' total compensation.

Any item that can store a value can be used as a calculated item by setting its required property values.

Calculation Modes

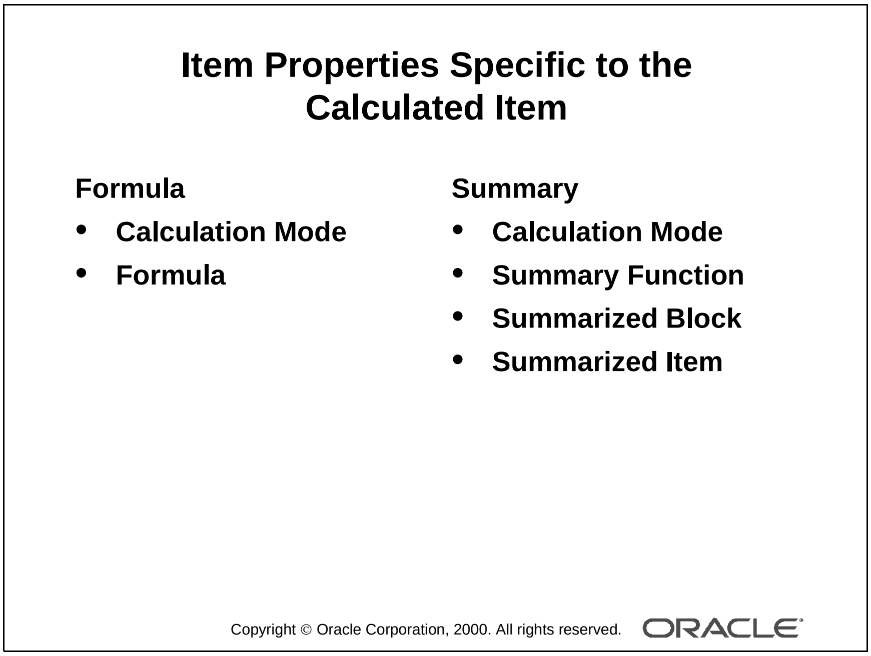
Calculations can be expressed as a formula or as a summary of all items in a block. Form Builder supports the following calculation modes:

Calculation Mode	**Description**
Formula	The calculated item value is the result of a horizontal calculation involving one or more bind variables, such as form items, global variables, and parameters
Summary	The calculated item value is a vertical calculation involving the values of a single item over all the rows within a single block

Note: A calculated item is read only. End users cannot insert or modify calculate items. You should, therefore, generally use display items as calculated items.

Instructor Note

Point out to the students that, unlike the rest of the items covered so far in this lesson, there is no Item Type property value called Calculated Item. The calculation-specific properties of an item make it a calculated item. Both text as well as display items support calculated items.



Item Properties Specific to the Calculated Item

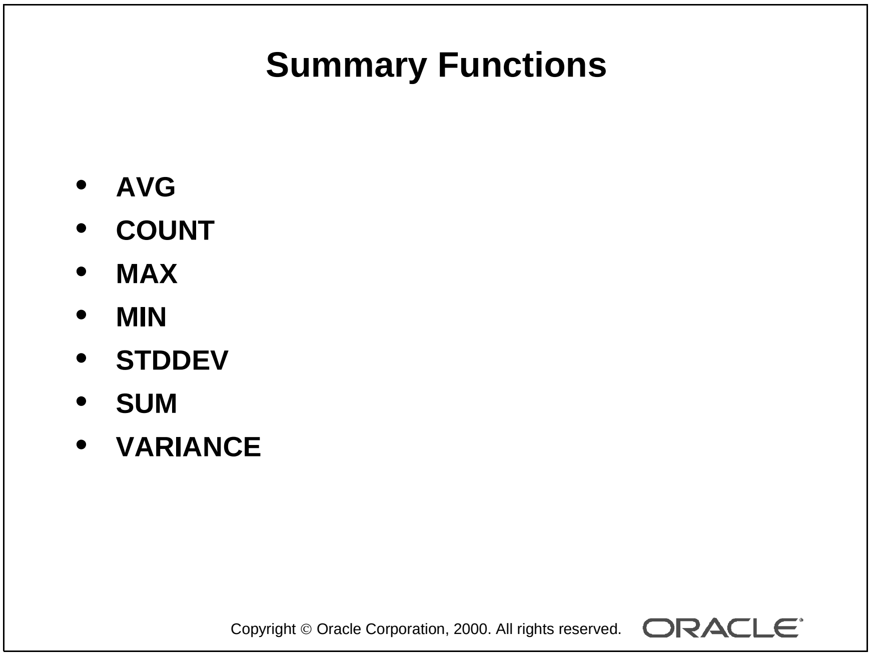
Formula

- **Calculation Mode**
- **Formula**

Summary

- **Calculation Mode**
- **Summary Function**
- **Summarized Block**
- **Summarized Item**

Copyright © Oracle Corporation, 2000. All rights reserved.



Summary Functions

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **STDDEV**
- **SUM**
- **VARIANCE**

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Calculated Item

A calculated item can be created by:

- Setting the calculation specific properties of any existing item that can store a value
- Creating a new item in the Layout Editor and setting its calculation specific properties
- Using the Create icon in the Object Navigator and setting its calculation specific properties

Item Properties Specific to the Calculated Item

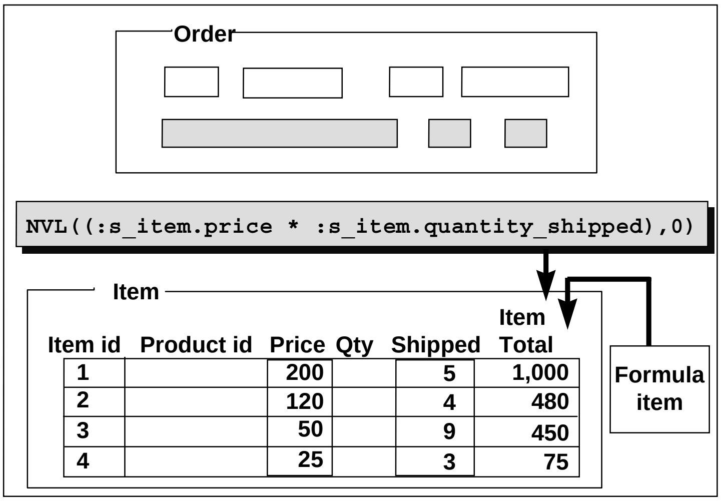
Property	Function
Calculation Mode	Specifies the method of computing the calculated item value (Choose from None, Formula, and Summary.)
Formula	Specifies a single PL/SQL expression that determines the calculated item value (The expression can compute a value or call a subprogram.)
Summary Function	Specifies the type of summary function to be performed on the calculated item (discussed later in this lesson).
Summarized Block	Specifies the block over which all rows will be summarized in order to assign a value to the calculated item; required if the item Calculation Mode property value is set to Summary
Summarized Item	Specifies the item whose value is summarized in order to be assign a value to the calculated item; required if the item Calculation Mode property value is set to Summary

Summary Functions

You can use the standard SQL aggregate functions for summary items:

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE

Item Based on a Formula



```
NVL((:s_item.price * :s_item.quantity_shipped),0)
```

Item id	Product id	Price	Qty	Shipped	Item Total
1		200		5	1,000
2		120		4	480
3		50		9	450
4		25		3	75

Copyright © Oracle Corporation, 2000. All rights reserved.

Rules for Formula Items

- A formula item must not invoke restricted built-ins.
- A formula item cannot execute any DML statements.
- Do not terminate a PL/SQL expression with a semicolon.
- Do not enter a complete PL/SQL statement in assignment expressions.

Copyright © Oracle Corporation, 2000. All rights reserved.

How to Create an Item Based on a Formula

1. Create a new item in the Object Navigator.
2. Open the Property Palette of the item.
3. Set the Calculation Mode property to Formula.
4. Click More for the Formula property and enter the PL/SQL expression to define the formula.

Note: A formula item cannot be a database item because its value is computed by Form Builder, not queried from a database column.

Rules for Creating an Item Based on a Formula

- The formula (and any user-written subprogram that calls it) must not invoke any restricted built-ins.
- The formula (and any user-written subprogram that calls it) cannot execute any DML statements.
- Do not terminate the PL/SQL expression with a semicolon.
- If the PL/SQL expression involves an assignment, do not enter the complete PL/SQL statement. Form Builder assigns the actual assignment code internally.

Example

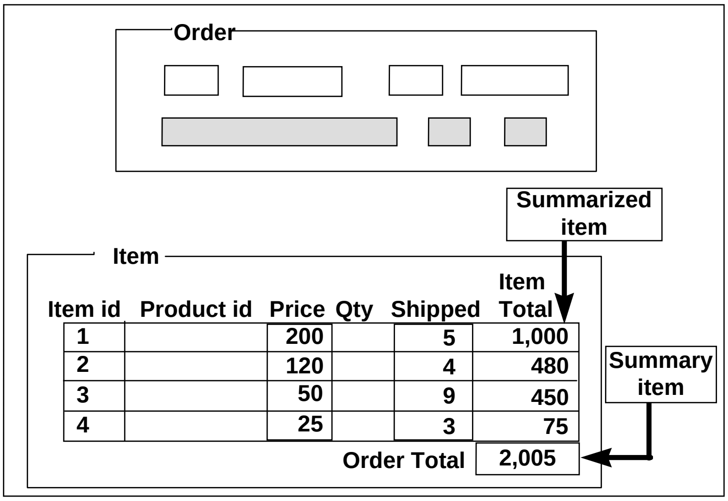
If you set the Formula property to:

```
NVL(:s_emp.salary,0) * NVL(:s_emp.commission_pct,0)
```

Form Builder will internally convert this expression into a complete statement as:

```
:s_emp.gross_comp :=
(NVL(:s_emp.salary,0) * NVL(:s_emp.commission_pct,0));
```

Item Based on a Summary



Item id	Product id	Price	Qty	Shipped	Item Total
1		200		5	1,000
2		120		4	480
3		50		9	450
4		25		3	75
				Order Total	2,005

Copyright © Oracle Corporation, 2000. All rights reserved.

Rules for Summary Items

- **Summary item must reside in:**
  - **The same block as the summarized item**
  - **A control block with Single Record property set to Yes**
- **Summarized item must reside in:**
  - **A data block with Query All Records property or Precompute Summaries property set to Yes**
  - **A control block**
- **Datatype of summary item must be Number, unless using MAX or MIN**

Copyright © Oracle Corporation, 2000. All rights reserved.

How to Create an Item Based on a Summary

1. Create a new item in the Object Navigator.
2. Open the Property Palette of an item.
3. Set the Calculation Mode property to Summary.
4. Select the required function from the Summary Function pop-up list.
5. From the Summarized Block pop-up list, select a block over which all rows will be summarized.
6. From the Summarized Item pop-up list, select an item to be summarized.

Note: A *summary item* is the calculated item to which you assign a value.

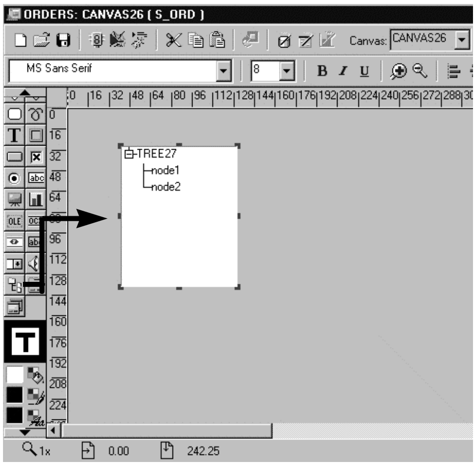
A *summarized item* is the item whose values are summarized and then assigned to the summary item.

Rules for Creating an Item Based on a Summary

- The summary item must reside in the same block as the summarized item, or in a control block whose Single Record property is set to Yes.
- The summarized item must reside in a control block, or in a data block whose Query All Records property or the Precompute Summaries property is set to Yes.

  Note: This ensures that records fetched in the block and the summarized value are consistent. Otherwise, another user may possibly update a record that has not been fetched yet.
- Set the Data Type property for a summary item to Number, unless the summary function is Max or Min, in which case the datatype must mirror that of its associated summarized item.
- If the summarized item values are based on a formula, the summarized item must reside in a block whose Query All Records property is set to Yes.

Creating a Hierarchical Tree



Copyright © Oracle Corporation, 2000. All rights reserved.

Hierarchical Tree Properties

- **Allow empty branches**
- **Multi selection**
- **Show lines**
- **Show symbols**
- **Record group**
- **Data query**

Copyright © Oracle Corporation, 2000. All rights reserved.

Creating a Hierarchical Tree Item

A hierarchical tree is an item that displays data in the form of a standard navigator.

How to Create a Hierarchical Tree Item

To create a hierarchical tree item, do one of the following:

- In the Layout Editor:
  - Click the Hierarchical Tree icon.
  - Click and drag the mouse on the canvas to create the hierarchical tree object.
  - Set other hierarchical tree-related properties as required.
- In the Object Navigator:
  - Create a new item by using the Create icon.
  - Open the item’s Property Palette and set the Item Type property to Hierarchical Tree.
  - Set other hierarchical tree related properties as required.

Hierarchical Tree Properties

Hierarchical Tree Properties include:

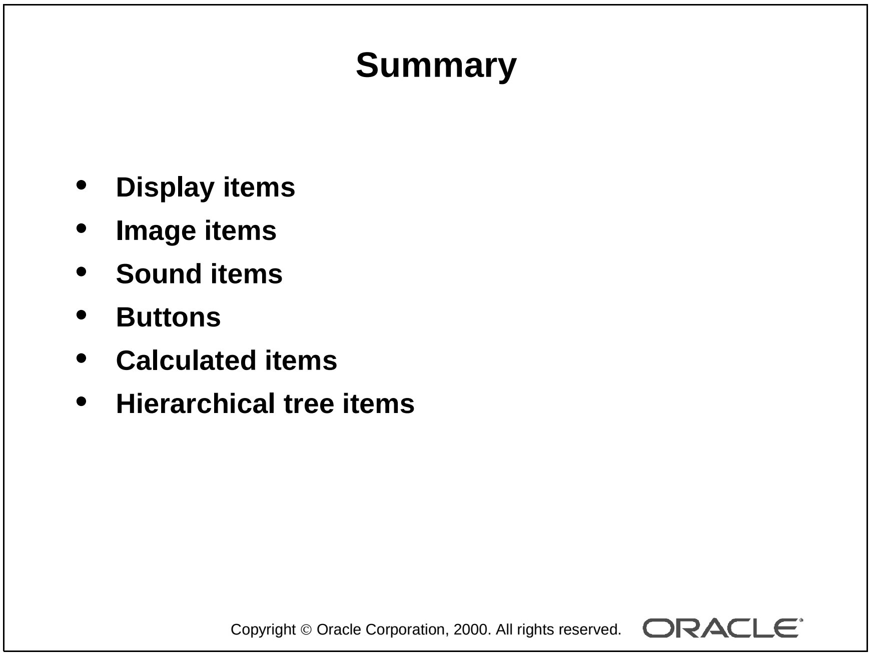
- Item Type
- Allow Empty Branches
- Multi Selection
- Show Lines
- Show Symbols
- Record Group
- Data Query

Several new built-ins are available to manipulate hierarchical trees. These are discussed in Lesson 15.

Note: A hierarchical tree must be the only item in the data block.

Instructor Note

Lesson 15 discusses how to implement a hierarchical tree in a form.



Summary

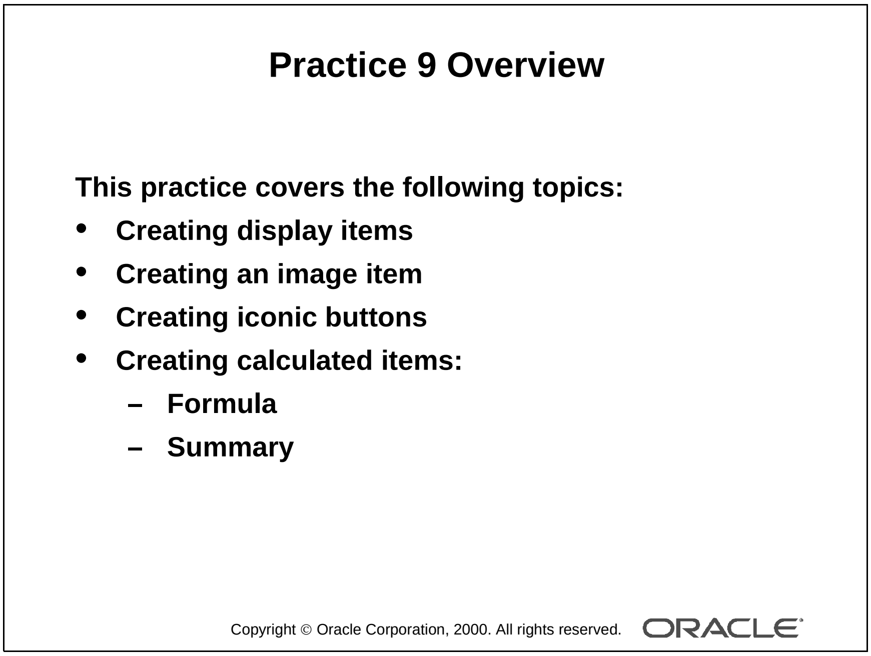
- Display items
- Image items
- Sound items
- Buttons
- Calculated items
- Hierarchical tree items

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Summary

In this lesson, you should have learned that:

- Display items display graphics or conditional text.
- Image items store and display vector or scanned bitmapped images.
- Sound items play and record sound data.
- Push Buttons initiate an action.
- Calculated items base item values on calculations. Calculations can be expressed in one of the following modes:
  - Formula
  - Summary
- Hierarchical trees display information in an Object Navigator style display.



Practice 9 Overview

This practice covers the following topics:

- **Creating display items**
- **Creating an image item**
- **Creating iconic buttons**
- **Creating calculated items:**
  - **Formula**
  - **Summary**

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

For solutions to this practice, see Practice 9 in Appendix A, “Practice Solutions.”

Practice 9 Overview

In this practice session, you will add several items in the CUSTOMERS and ORDERS forms: display items, image item, push buttons, and calculated items.

- In the ORDERS form, create two display items in the S_ITEM block.
- Create an image item in the S_ITEM block.
- In the ORDERS form, create an iconic button in the control block.
- In the ORDERS form, base the Item_Total item in the S_ITEM block on a formula. Create a control item in the same block. Base this item value on a summary that displays the total value of an order.
- In the CUSTOMERS form, create an iconic button in the CONTROL block.
- Save and run the ORDERS and CUSTOMERS forms.

Practice 9

1 In the S_ITEM block of the ORDG*XX* form, create a display item called Description. Set the Prompt property to Description and display the prompt above the item.

2 Create a single-record image item called Product_Image in the S_ITEM block of the ORDG*XX* form.

3 Create another display item, Image_Description, in the S_ITEM block. This should synchronize with the Description item. Set the Maximum Length property to the same value as the Description item.

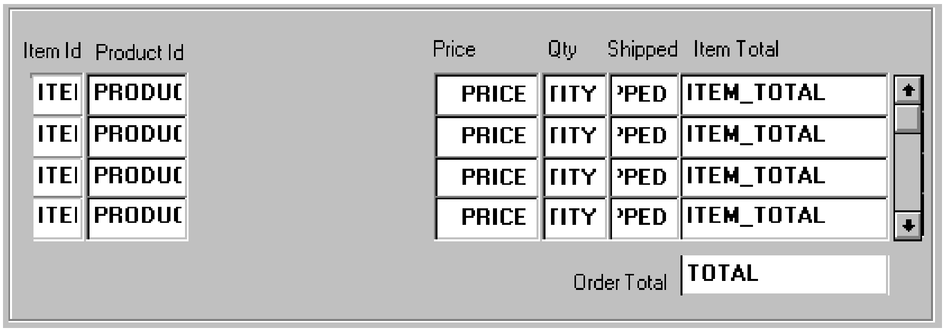
4 In the CONTROL block of the ORDG*XX* form, create an iconic button called Product_LOV_Button. Use the `list` file (do not include the `.ico` or `.gif` extension). Set both the Keyboard Navigable property and the Mouse Navigate property to No.

5 To display item total information, set the following properties for the Item_Total item in the S_ITEM block:

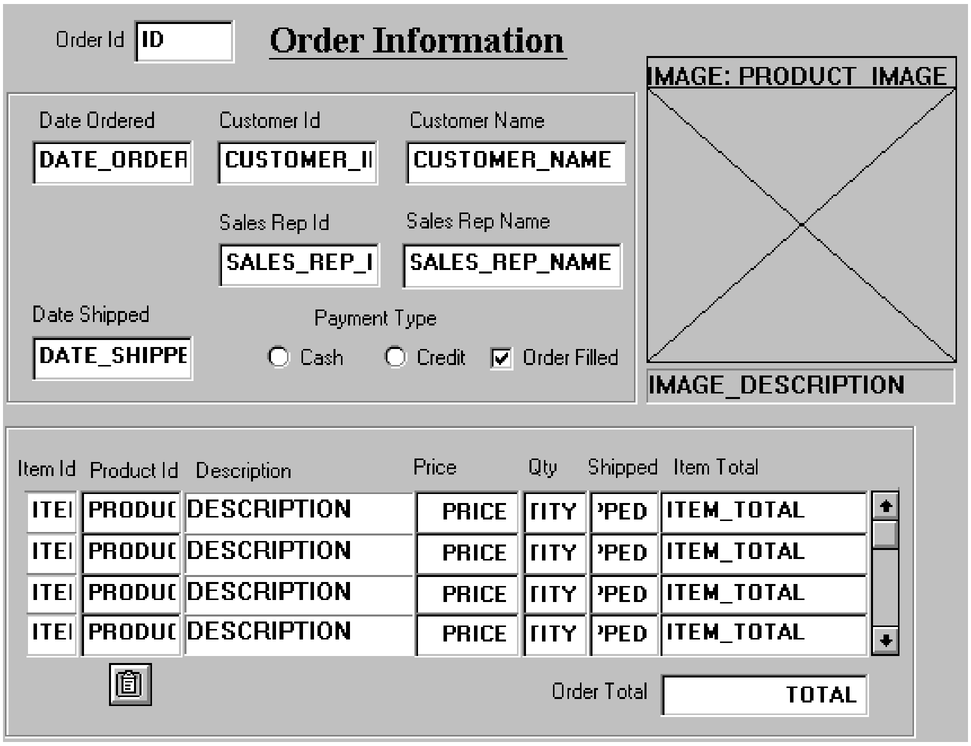
- Set the Justification property to right.
- Set the Calculation Mode property to Formula.
- Set the Formula property to :S_ITEM.quantity_shipped * :S_ITEM.price.
- Set the Keyboard Navigable property to No.

6 To display the total of the item totals create a new nondatabase item in the S_ITEM block.

- Set the position, size and prompt properties according to the screenshot.
- Set the format mask property to 9G999G990D99.
- Set the Justification property to right.
- Set the Number of Items Displayed property to 1.
- Make S_ITEM.total a summary item and display summaries of the item_total values in the S_ITEM block. Ensure that you have to set the Query All Records property to Yes for the S_ITEM block.
- Set the Keyboard Navigable property to No.



7 Save, compile, and run the forms to test the changes. Change the window size if necessary.



8 Perform a query in the ORDG*XX* form to ensure that the new items do not cause an error. Did you remember to switch off the Database Item property for items that do not correspond to columns in the base table?

9 Create an iconic button similar to the one created in question 4, in the CONTROL block of form CUSTG*XX*. Use the `list` file (do not include the `.ico` or `.gif` extension). Name the push button Sales_Rep_Lov_Button, and place it next to Sales_Rep_ID.

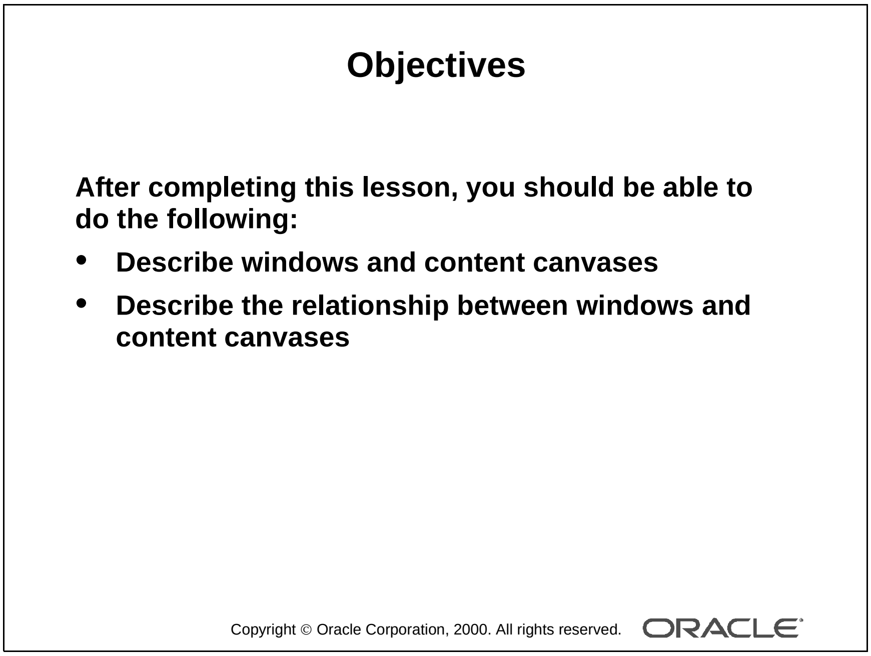
10 Save and compile the form.

Deploy the form on the Web to test the changes.

Note: When you test the form on the Web, the iconic button will have a different image than the same iconic button in the client-server environment. This is because the image stored in `list.gif` is different from the image stored in `list.ico`.

10

Creating Windows and Content Canvases

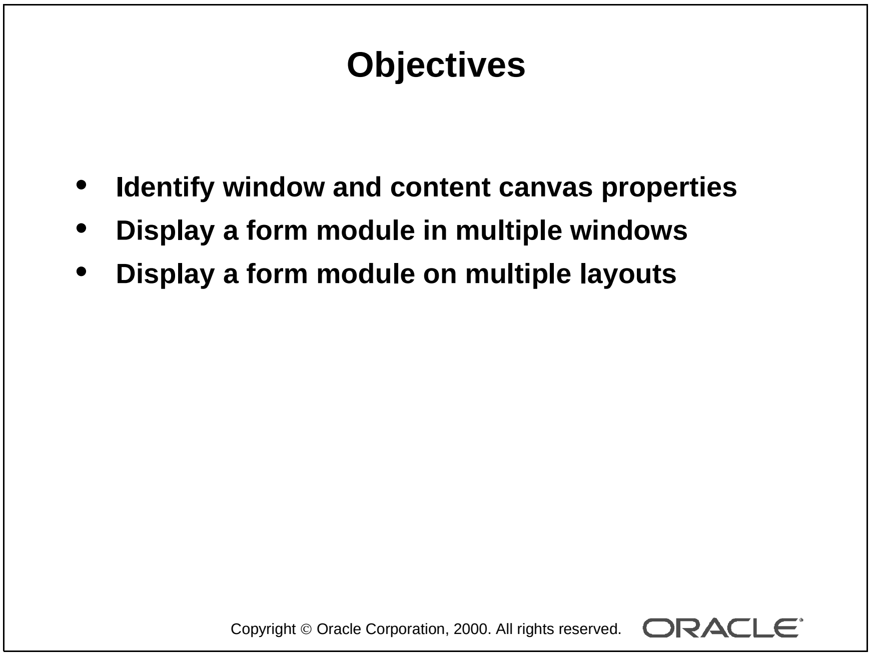


Objectives

After completing this lesson, you should be able to do the following:

- Describe windows and content canvases
- Describe the relationship between windows and content canvases

Copyright © Oracle Corporation, 2000. All rights reserved.



Objectives

- Identify window and content canvas properties
- Display a form module in multiple windows
- Display a form module on multiple layouts

Copyright © Oracle Corporation, 2000. All rights reserved.

Schedule

Topic	Timing
Lecture	30 minutes
Practice	20 minutes
Total	50 minutes

Introduction

Overview

With Oracle Forms Developer you can take advantage of the GUI environment by displaying a form module across several canvases and in multiple windows. This lesson familiarizes you with the window object and the default canvas type, the content canvas.

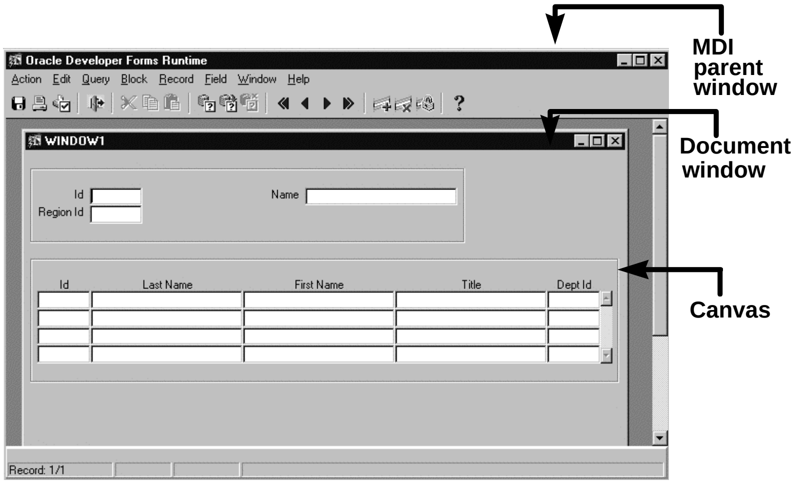
Windows and Canvases

- Window: Container for Form Builder visual objects
- Canvas: Surface on which you “paint” visual objects

  To see a canvas and its objects, display the canvas in a window.

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Window, Canvas, and Viewport



Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Instructor Note

You can demonstrate ownership and visual view at this point.

Windows and Content Canvases

With Form Builder you can display an application in multiple windows by using its display objects—windows and canvases.

What Is a Window?

A *window* is a container for all visual objects that make up a Form Builder application. It is similar to an empty picture frame. The window manager provides the controls for the window that enable such functionality as scrolling, moving, and resizing. You can minimize a window.

A single form may include several windows.

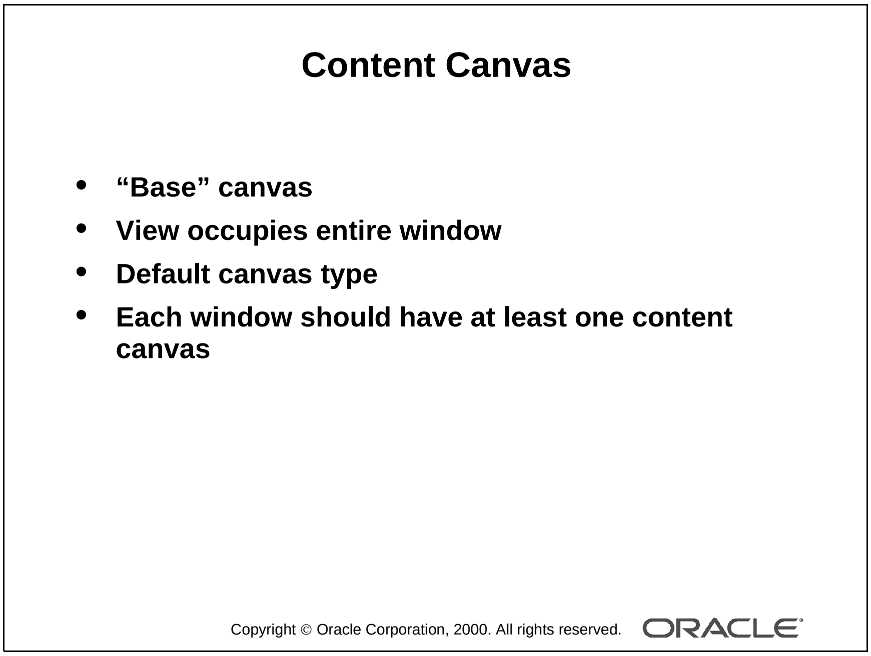
What Is a Canvas?

A *canvas* is a surface inside a window container on which you place visual objects such as interface items and graphics. It is similar to the canvas upon which a picture is painted. To see a canvas and its contents at run time, you must display it in a window. A canvas always displays in the window to which it is assigned.

Note: Each item in a form must refer to no more than one canvas. An item displays on the canvas to which it is assigned, through its Canvas property. Recall that if the Canvas property for an item is left unspecified, that item is said to be a Null-canvas item and will not display at runtime.

What Is a Viewport?

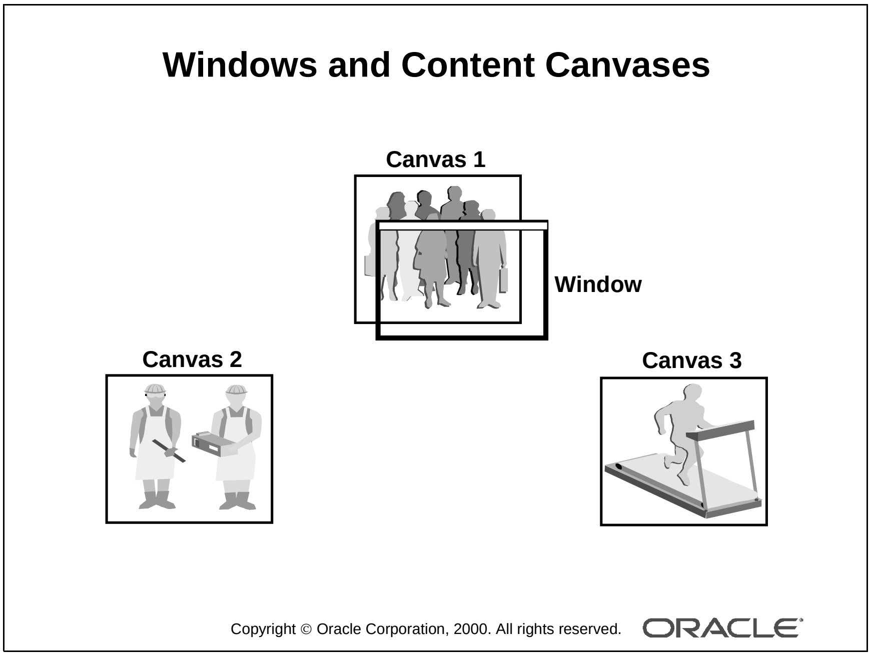
A *viewport* is an attribute of a canvas. It is effectively the visible portion of, or view onto, the canvas.



Content Canvas

- **“Base” canvas**
- **View occupies entire window**
- **Default canvas type**
- **Each window should have at least one content canvas**

Copyright © Oracle Corporation, 2000. All rights reserved.



Windows and Content Canvases

Copyright © Oracle Corporation, 2000. All rights reserved.

Note

At run time, only one content canvas can display in a window at a time. However, you can assign multiple content canvases to a window.

What Is a Content Canvas?

Form Builder offers different types of canvases. A *content canvas* is the base canvas that occupies the entire content pane of the window in which it displays. The content canvas is the default canvas type. Most canvases are content canvases.

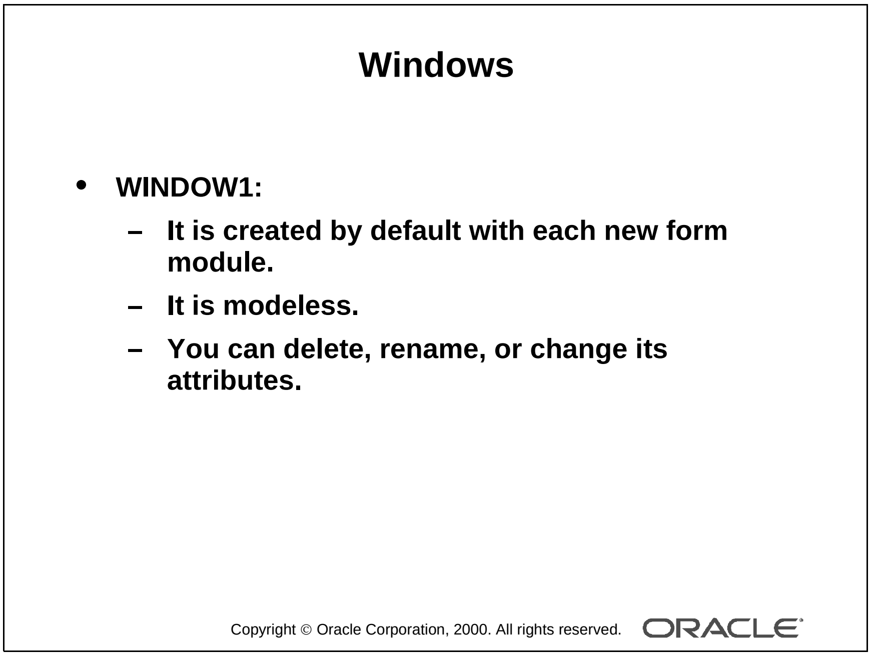
The Relationship Between Windows and Content Canvases

You must create at least one content canvas for each window in your application. When you run a form, only one content canvas can display in a window at a time, even though more than one content canvas can be assigned to the same window at design time.

At run time, a content canvas always completely fills its window. As the user resizes the window, Form Builder resizes the canvas automatically. If the window is too small to show all items on the canvas, Form Builder automatically scrolls the canvas to bring the current item into view.

Instructor Note

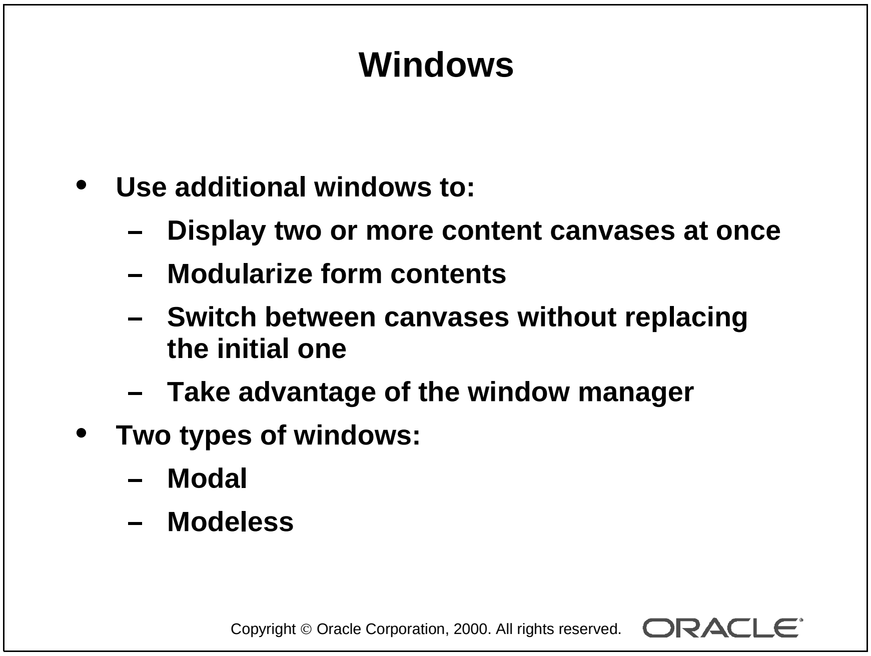
Point out to the students that the rest of the canvas types will be covered in the next lesson.



Windows

- **WINDOW1:**
  - **It is created by default with each new form module.**
  - **It is modeless.**
  - **You can delete, rename, or change its attributes.**

Copyright © Oracle Corporation, 2000. All rights reserved.



Windows

- **Use additional windows to:**
  - **Display two or more content canvases at once**
  - **Modularize form contents**
  - **Switch between canvases without replacing the initial one**
  - **Take advantage of the window manager**
- **Two types of windows:**
  - **Modal**
  - **Modeless**

Copyright © Oracle Corporation, 2000. All rights reserved.

Displaying a Form Module in Multiple Windows

When you create a new form module, Form Builder creates a new window implicitly. Thus, each new form module has one predefined window, which is called WINDOW1. You can delete or rename WINDOW1, or change its attributes.

Uses and Benefits of a New Window

You can create additional windows in which to display your form application. A new or second window provides the ability to do the following:

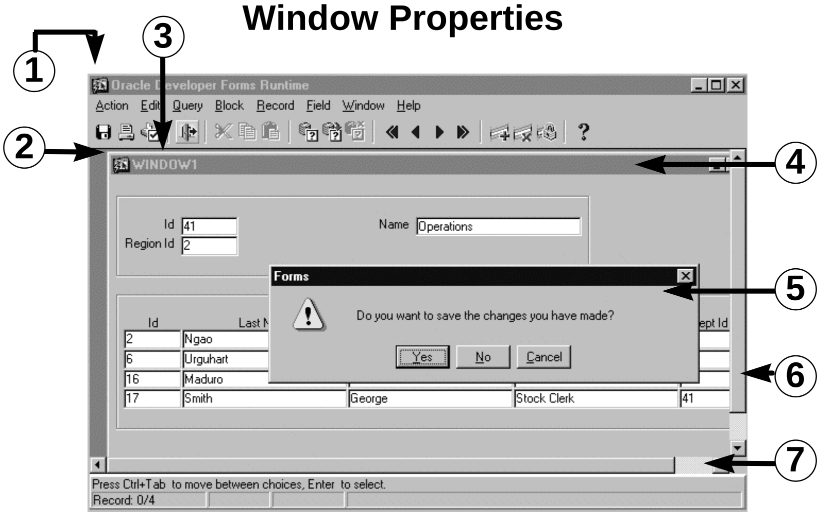
- Display two or more content canvases at once.
- Modularize the form contents.
- Switch between canvases without replacing the initial one.
- Take advantage of window manager functionality such as minimizing.

Window Types

You can create two different window types: modal and modeless.

- A *modal window* is a restricted window that the user must respond to before moving the input focus to another window. Modal windows:
  - Must be dismissed before control can be returned to a modeless window
  - Become active as soon as they display
  - Require a means of exit or dismissal
- A *modeless window* is an unrestricted window that the user can exit freely. Modeless windows:
  - Can display many at once
  - Are not necessarily active when displayed
  - Are the default window type

Window Properties


1
2
3
4
5
6
7
Oracle Developer Forms Runtime
Action Edit Query Block Record Field Window Help
WINDOW1
Forms
Yes
No
Cancel

Id 41 Name Operations
Region Id 2

Do you want to save the changes you have made?

Id	Last N			ept Id
2	Ngao			
6	Urguhart			
16	Maduro			
17	Smith	George	Stock Clerk	41

Press Ctrl+Tab to move between choices, Enter to select.
Record: 0/4

Copyright © Oracle Corporation, 2000. All rights reserved.

1	MDI parent window
2	X/Y position
3	Title
4	Document window
5	Dialog window
6	Show vertical scroll bar
7	Show horizontal scroll bar

Window Properties

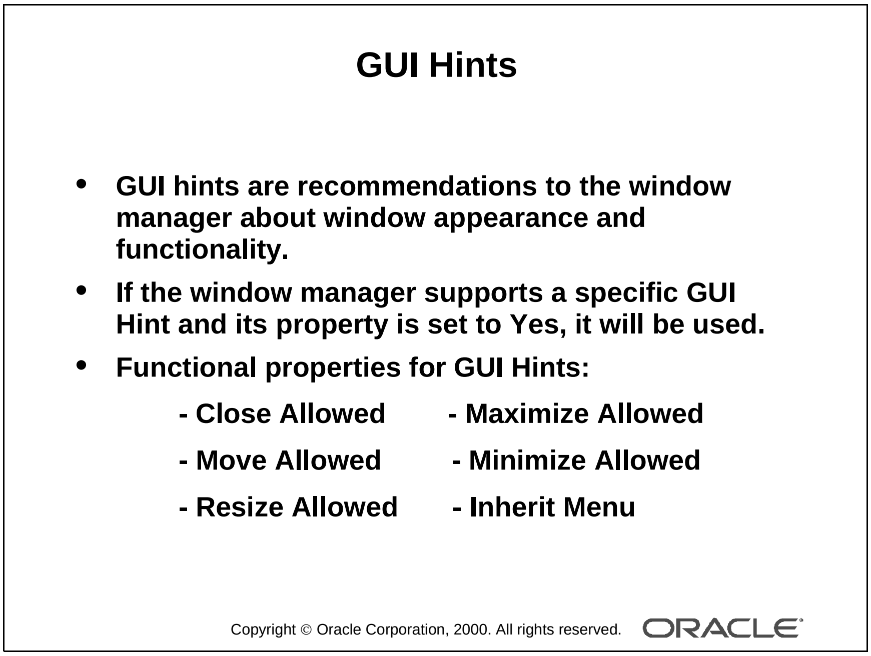
Physical Property	Function
X Position	Determines the X coordinate for the window
Y Position	Determines the Y coordinate for the window
Width	Determines the width of the window
Height	Determines the height of the window
Bevel	Determines how the window border displays
Show Horizontal Scrollbar	Determines whether a horizontal scroll bar displays in the window
Show Vertical Scrollbar	Determines whether a vertical scroll bar displays in the window

Functional Property	Function
Title	Specifies a window title to appear in the title bar
Primary Canvas	Specifies the name of the canvas to display in this window when it is invoked programmatically
Window Style	Determines whether the window style is Document or Dialog (Document style windows are fixed and always remain within the application window frame. Dialog style windows are free floating and can be moved outside the application window frame.)
Modal	Determines whether the window is modal (requires user response) or modeless (does not require user response)
Hide on Exit	Specifies whether a modeless window is hidden automatically when the end user navigates to an item in another window
Icon Filename	Specifies the icon resource name that depicts the minimized window

Note: If you do not specify a window title, Form Builder uses the window object name specified in the Name property for the title.

The canvas you choose as the primary canvas must be a content canvas.

The X and Y Position (0,0) of a window is relative to the top left corner of the screen when you set the Window Style to dialog. If you set the Window Style to document, the X and Y Position (0,0) is relative to the top-left corner of the MDI window.



GUI Hints

- **GUI hints are recommendations to the window manager about window appearance and functionality.**
- **If the window manager supports a specific GUI Hint and its property is set to Yes, it will be used.**
- **Functional properties for GUI Hints:**

- Close Allowed	**- Maximize Allowed**
- Move Allowed	**- Minimize Allowed**
- Resize Allowed	**- Inherit Menu**

Copyright © Oracle Corporation, 2000. All rights reserved.

Instructor Note

The Resize Allowed property can be set to prevent an end user from resizing the window, but it does not prevent you from resizing the window programmatically.

Demonstration Using the `ordwk09.fmb` file, create a new window in the Orders form. Show that the new window does not display at run time. Explain that the new window will not display until it is specified in the Window property of a canvas.

What Are GUI Hints?

GUI Hints are recommendations to the window manager about the window appearance and functionality. There are certain properties under the Functional group that enable you to make these recommendations. If the current window manager supports the specific GUI Hint property and it is set to Yes, then Form Builder uses it. However if the window manager does not support the property, Form Builder ignores it.

Functional Property for GUI Hints	**Function**
Close Allowed	Enables the mechanism for closing the window, as provided by the window manager-specific Close command (Form Builder responds to user attempts to close the window by firing a WHEN-WINDOW-CLOSED trigger to actually close it.)
Move Allowed	Determines whether the user can move the window by using the means provided by the window manager
Resize Allowed	Determines whether the user can resize the window at run time
Maximize Allowed	Determines whether the user can resize the window by using the zooming capabilities of the window manager
Minimize Allowed	Determines whether the user can iconify and minimize the window
Inherit Menu	Determines whether the window displays the current form menu

Note: The Minimize Allowed property must be set to Yes in order for Icon Filename to be valid.

How to Create a New Window

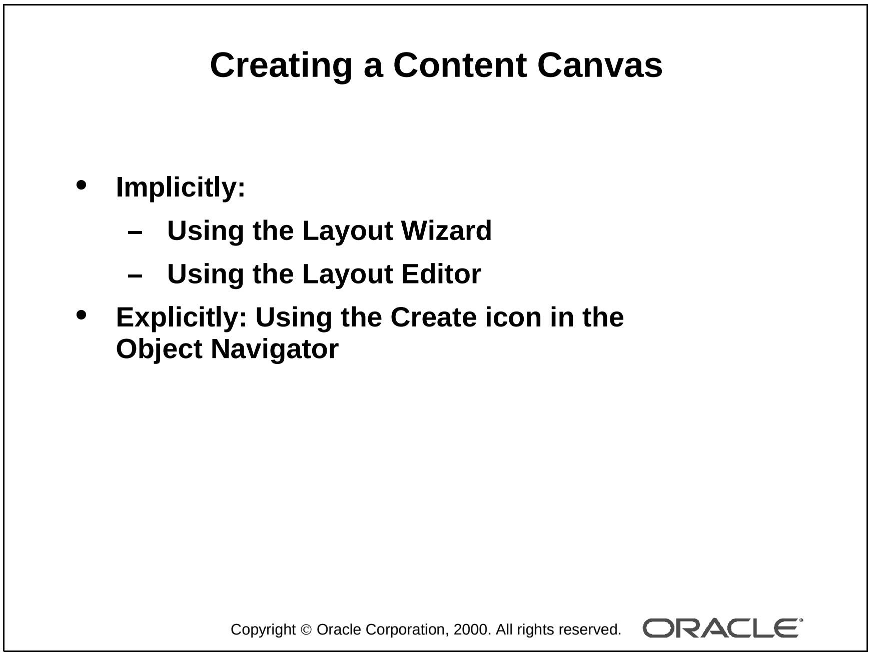
1. Click the Windows node in the Object Navigator.
2. Click the Create icon.

   A new window entry displays, with a default name of WINDOW*XX*.
3. If the Property Palette is not already displayed, double-click the window icon to the left of the new window entry.
4. Set the window properties according to your requirements (as described in the tables, earlier in this lesson).

Note: For your new window to display, you must specify its name in the Window property of at least one canvas.

Web Design Tip

To display a console to end users, set the form-level property Console Window to the window in which you want to display the console.



Creating a Content Canvas

- **Implicitly:**
  - **Using the Layout Wizard**
  - **Using the Layout Editor**
- **Explicitly: Using the Create icon in the Object Navigator**

Copyright © Oracle Corporation, 2000. All rights reserved.

Displaying a Form Module on Multiple Layouts

You can have more than one content canvas in your form application. However, remember that only one content canvas can display in a window at one time. To display more than one content canvas at the same time, you can assign each content canvas to a different window.

Now you can display the form module on multiple layouts or surfaces.

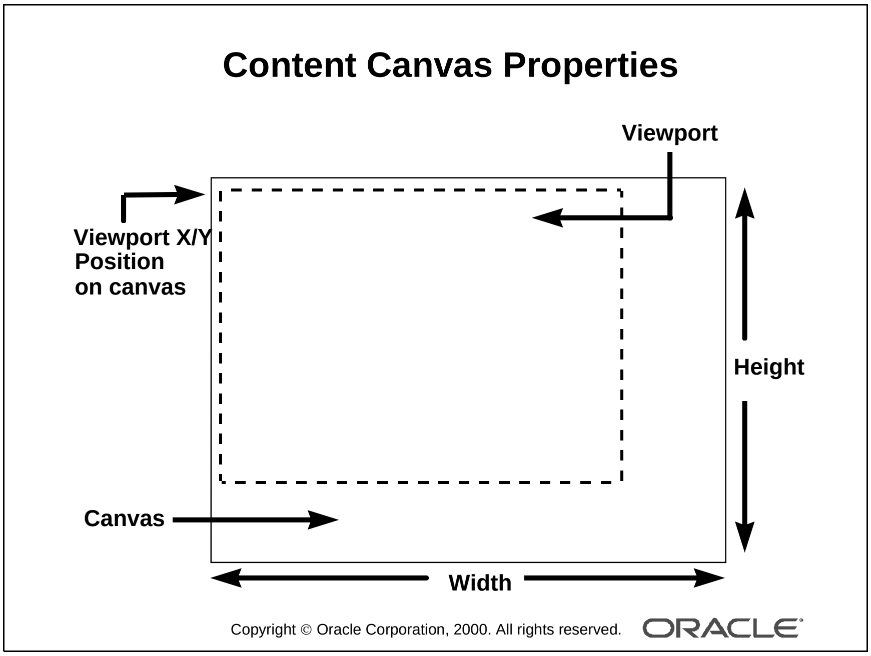
Creating a New Content Canvas Implicitly

There are two ways of implicitly creating a new content canvas:

- Layout Wizard: When you use the Layout Wizard to arrange data block items on a canvas, the wizard enables you to select a new canvas on its Canvas page. In this case, the wizard creates a new canvas with a default name of CANVAS*XX*.
- Layout Editor: When there are no canvases in a form module and you invoke the Layout Editor; Form Builder creates a default canvas on which you can place items.

Creating a New Content Canvas Explicitly

You can create a new content canvas explicitly by using the Create icon in the Object Navigator.



Content Canvas Properties

Viewport

Viewport X/Y Position on canvas

Height

Canvas

Width

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Instructor Note

Demonstration Display the contents of the S_INVENTORY data block in the new window created in the previous demonstration.

Also, since Release 6, the Show View option in the View menu of the Layout Editor displays the size of the window for a content canvas.

Content Canvas Specific Properties

General Property	Function
Canvas Type	Specifies the type of canvas (For a content canvas, this property should be set to Content.)

Physical Property	Function
Window	Specifies the window in which the canvas will be displayed
Viewport X Position on Canvas	Specifies the X coordinate of the top-left corner of the view relative to the upper-left corner of the canvas
Viewport Y Position on Canvas	Specifies the Y coordinate of the top-left corner of the view relative to the upper-left corner of the canvas
Width	Specifies the width of the canvas
Height	Specifies the height of the canvas
Bevel	Specifies a sculpted effect canvas border

Functional Property	Function
Raise on Entry	Determines whether the canvas is always brought to the front of the window when the user navigates to an item on this canvas (Use this property when the canvas is displayed in the same window with other types of canvases.)

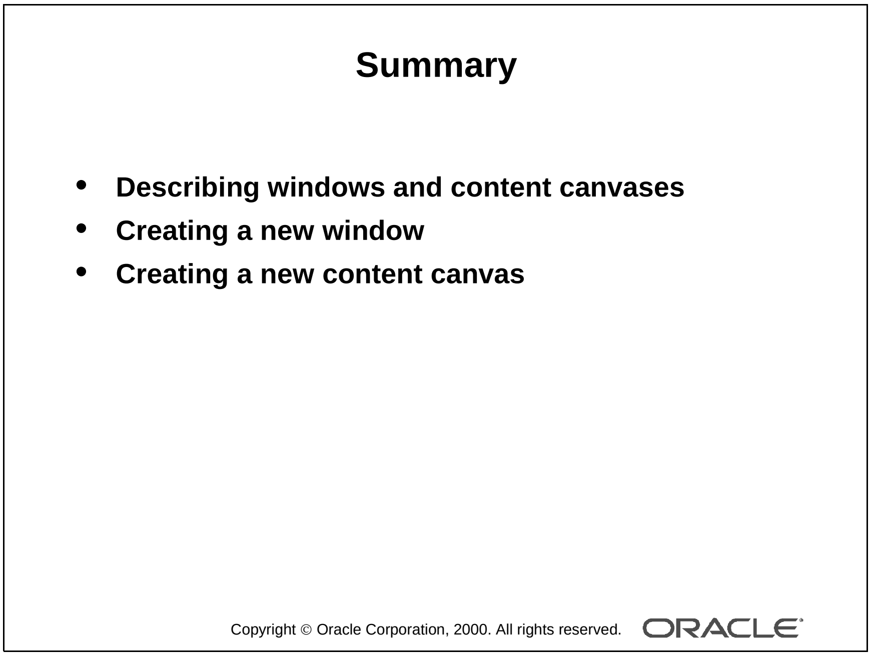
Note: For a canvas to display at run time, its Window property must be specified.

How to Create a New Content Canvas

1. Click the Canvases node in the Object Navigator.
2. Click the Create icon.

   A new canvas entry displays with a default name of CANVAS*XX*.
3. If the Property Palette is not already displayed, click the new canvas entry and select Tools—>Property Palette.
4. Set the canvas properties that are described in the above tables according to your requirements.

Note: Double-clicking the icon for a canvas in the Object Navigator will invoke the Layout Editor instead of the Property Palette.



Summary

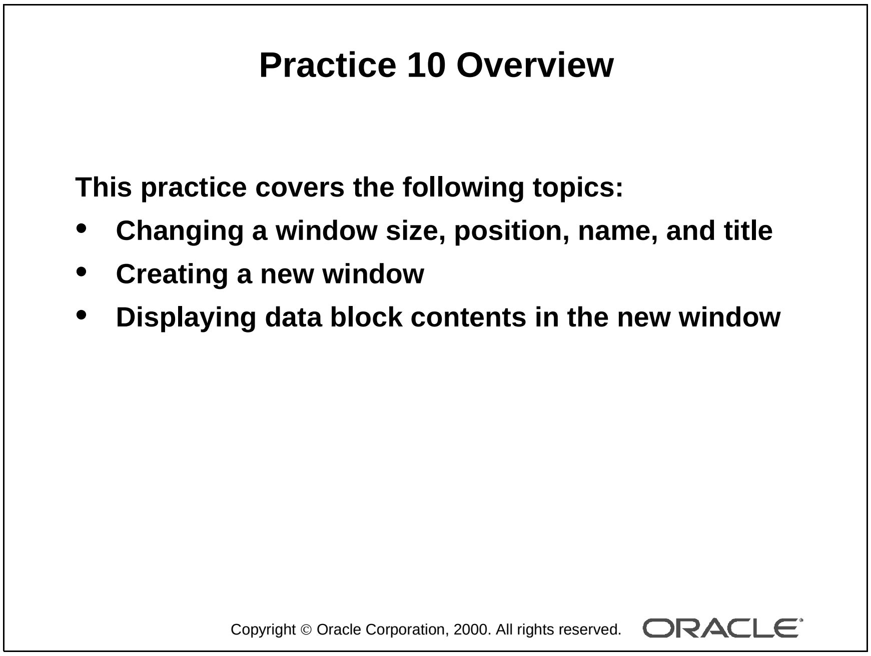
- **Describing windows and content canvases**
- **Creating a new window**
- **Creating a new content canvas**

Copyright © Oracle Corporation, 2000. All rights reserved.

Summary

In this lesson, you should have learned:

- About the relationship between windows and content canvases
- How to create a new window
- How to create a new content canvas



Practice 10 Overview

This practice covers the following topics:

- **Changing a window size, position, name, and title**
- **Creating a new window**
- **Displaying data block contents in the new window**

Copyright © Oracle Corporation, 2000. All rights reserved. ORACLE

Note

For solutions to this practice, see Practice 10 in Appendix A, “Practice Solutions.”

Practice 10 Overview

In this practice session, you will customize windows in your form modules. You will resize the windows to make them more suitable for presenting canvas contents. You will also create a new window to display the contents of the S_INVENTORY block.

- Change the size and position of the window in the CUSTOMERS form. Change its name and title. Save and run the form.
- Modify the name and title of the window in the ORDERS form.
- Create a new window in the ORDERS form. Make sure the contents of the S_INVENTORY block display in this window. Save and run the form.

Practice 10

1 Modify the window in the CUSTG*XX* form. Change the name of the window to WIN_CUSTOMER, and change its title to Customer Information. Check that the size and position are suitable.

2 Save, compile, and run the form to test the changes.

3 Modify the window in the ORDG*XX* form. Ensure that the window is called WIN_ORDER. Also change its title to Orders and Items.

4 In the ORDG*XX* form, create a new window called WIN_INVENTORY suitable for displaying the CV_INVENTORY canvas. Use the rulers in the Layout Editor to help you plan the height and width of the window. Set the window title to Stock Levels and the Hide on Exit property to Yes. Place the new window in a suitable position relative to WIN_ORDER.

5 Associate the CV_INVENTORY canvas with the window WIN_INVENTORY. Compile the form. Deploy the form on the Web and ensure that the S_INVENTORY block displays in WIN_INVENTORY when you navigate to this block.

6 Save the form.