

Oracle Fusion Middleware 11g: Build Applications with Oracle Forms

Duration: 5 Days

What you will learn

This Oracle Fusion Middleware 11g: Build Applications with Oracle Forms training explores building Oracle Forms Builder 11g. Expert instructors will teach you how to create high-performance applications for the Web that are also scalable.

Learn To:

Use Oracle Forms Builder 11g.

Enhance applications with various GUI controls.

Add functionality to applications by writing triggers.

Use the Forms Debugger to troubleshoot applications.

Validate user input and display meaningful error messages.

Use WebUtil to interact with client computers.

Validate user input, control navigation and display meaningful messages to users.

Use WebUtil to interact with the client machine and pass values from one form to another.

Replace or supplement default transaction processing.

Build a Sample Forms Application

You'll also learn to build a sample Forms application for an order entry system using a variety of GUI controls. Expert Oracle University instructors will show you how to enhance the appearance and functionality of the basic form by using PL/SQL trigger, JavaBeans, and Pluggable Java Components.

Note:

Emphasis is placed on designing objects and code for reuse.

Audience

Application Developers

Developer

Forms Developer

PL/SQL Developer

Support Engineer

Technical Consultant

Related Training

Required Prerequisites

Experience with SQL and PL/SQL basics

Suggested Prerequisites

Experience in advanced SQL & PL/SQL

Oracle Database 11g: Advanced PL/SQL

Course Objectives

Create form modules, including components for database interaction and GUI controls

Display form modules in multiple windows and use a variety of layout styles

Test form modules in a Web browser

Debug form modules in a 3-tier environment

Implement triggers to enhance form functionality

Reuse objects and code

Link one form module to another

Course Topics

Running a Forms Application

Running a Form

Identifying the Data Elements

Navigating a Forms Application

Using the Modes of Operation

Querying Data

Inserting, Updating, and Deleting Records

Saving Changes

Displaying Errors

Working in the Forms Builder Environment

Forms Builder Key Features

Forms Builder Components

Navigating the Forms Builder Interface

Forms Builder Module Object Hierarchy

Customizing Your Forms Builder Session

Forms Executables and Module Types

Defining Environment Variables

Testing a Form with the Run Form Button

Creating a Basic Form Module

Creating a New Form Module

Creating a New Data Block

Using Template Forms

Saving and Compiling a Form Module

Module Types and Storage Formats

Deploying a Form Module
Producing Documentation

Creating a Master-Detail Form

Creating Data Blocks with Relationships
Running a Master-Detail Form Module
Modifying the Structure of a Data Block
Modifying the Layout of a Data Block

Working Data Blocks and Frames

Managing Object Properties
Creating Visual Attributes
Controlling the Behavior and Appearance of Data Blocks
Controlling Frame Properties
Displaying Multiple Property Palettes
Setting Properties on Multiple Objects
Copying Properties
Creating Control Blocks

Working with Text Items

Creating a Text Item
Modifying the Appearance of a Text Item
Controlling the Data of a Text Item
Altering the Navigational Behavior of a Text Item
Enhancing the Relationship between Text Item and Database
Adding Functionality to a Text Item
Displaying Helpful Messages

Creating LOVs and Editors

LOVs and Record Groups
Creating an LOV Manually
Using the LOV Wizard to Create an LOV
Setting LOV Properties
LOV Column Mapping
Defining an Editor
Setting Editor Properties
Associating an Editor with a Text Item

Creating Additional Input Items

Input Items Overview
Creating a Check Box
Creating a List Item
Creating a Radio Group

Creating Noninput Items

Noninput Items Overview
Creating a Display Item
Creating an Image Item
Creating a Push Button
Creating a Calculated Item
Creating a Hierarchical Tree Item
Creating a Bean Area Item

Creating Windows and Content Canvases

Displaying a Form Module in Multiple Windows

Creating a New Window

Displaying a Form Module on Multiple Layouts

Creating a New Content Canvas

Working with Other Canvas Types

Overview of Canvas Types

Creating a Stacked Canvas

Creating a Toolbar

Creating a Tab Canvas

Producing and Debugging Triggers

Trigger Overview

Creating Triggers in Forms Builder

Specifying Execution Hierarchy

PL/SQL Editor Features

Using the Database Trigger Editor

Using Variables in Triggers

Adding Functionality with Built-in Subprograms

Using the Forms Debugger

Adding Functionality to Items

Coding Item Interaction Triggers

Interacting with Noninput Items

Displaying Run-Time Messages and Alerts

Built-Ins and Handling Errors

Controlling System Messages

The FORM_TRIGGER_FAILURE Exception

Triggers for Intercepting System Messages

Creating and Controlling Alerts

Handling Server Errors

Using Query Triggers

Query Processing Overview

SELECT Statements Issued During Query Processing

Setting WHERE and ORDER BY clauses and ONETIME_WHERE property

Writing Query Triggers

Query Array Processing

Coding Triggers for Enter-Query Mode

Overriding Default Query Processing

Obtaining Query Information at Run Time

Validating User Input

Validation Process

Controlling Validation by Using Properties

Controlling Validation by Using Triggers

Performing Client-Side Validation with PJC

Tracking Validation Status

Using Built-ins to Control When Validation Occurs

Controlling Navigation

- Using Object Properties to Control Navigation
- Writing Navigation Triggers
- Avoiding the Navigation Trap
- Using Navigation Built-Ins in Triggers

Overriding or Supplementing Transaction Processing

- Transaction Processing Overview
- Using Commit Triggers
- Testing the Results of Trigger DML
- DML Statements Issued during Commit Processing
- Overriding Default Transaction Processing
- Getting and Setting the Commit Status
- Implementing Array DML

Writing Flexible Code

- What is Flexible Code?
- Using System Variables for Flexible Coding
- Using Built-in Subprograms for Flexible Coding
- Copying and Subclassing Objects and Code
- Referencing Objects by Internal ID
- Referencing Items Indirectly

Sharing Objects and Code

- Benefits of Reusable Objects and Code
- Working with Property Classes
- Working with Object Groups
- Copying and Subclassing Objects and Code
- Working with Object Libraries
- Working with SmartClasses
- Reusing PL/SQL
- Working with PL/SQL Libraries

Using WebUtil to Interact with the Client

- Benefits of WebUtil
- Integrating WebUtil into a Form
- Interacting with the Client

Introducing Multiple Form Applications

- Multiple Form Applications Overview
- Starting Another Form Module
- Defining Multiple Form Functionality
- Sharing Data among Modules