

Oracle Forms Developer: Build Internet Applications

Duration: 50 Hours

What you will learn

Leverage your investment by taking advantage of web technologies to easily and quickly construct sophisticated database forms and business logic with minimal effort. This course focuses on teaching students to use Oracle Forms Developer (10.1.2.0.2) to rapidly build scalable, high-performance applications for the Internet.

In this course students build, test, debug, and deploy interactive Internet applications. Working in a graphical user interface (GUI) environment, they develop an order entry application from the ground up. This application incorporates several advanced features that provide a rich user experience while implementing business rules.

This course counts towards the Hands-on course requirement for the Oracle Forms Developer Certified Professional Certification. Only instructor-led inclass or instructor-led online formats of this course will meet the Certification Hands-on Requirement. Self Study CD-Rom and Knowledge Center courses are excellent study and reference tools but DO NOT meet the Hands-on Requirement for certification.

Learn To:

Customize forms with user input items such as check boxes, list items, radio groups, and Pluggable Java

Components Integrate Java into Forms applications by using JavaBeans

Control navigation, data access, validation, and transactions by creating event-related triggers Enable

Forms applications running on the Web to access files and applications on the client computer

Display Forms elements and data in multiple canvases and windows

Deploy Forms applications to the Web

Audience

Application Developers

Developer

Forms Developer

PL/SQL Developer

Support Engineer

Technical Consultant

Prerequisites

Required Prerequisites

Oracle Database : Advanced PL/SQL

Suggested Prerequisites

A good familiarity with Graphical User Interface (GUI)

Working experience with the Web browser

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
Reuse objects and code
Link one form module to another

Course Topics

Introducing Oracle Forms Developer and Forms Services

Grid Computing
Oracle Products
Oracle Application Server Architecture
Benefits and Components of Oracle Developer Suite
Running a Forms Developer Application
Working in the Forms Developer Environment

Creating Forms Modules

Creating a Basic Forms Module
Creating a Master-Detail Forms Module
Modifying the Data Block
Modifying the Layout

Working with Data Blocks and Frames

Using the Property Palette
Managing Object Properties
Creating and Using Visual Attributes
Controlling the Behavior and Appearance of Data
Blocks Controlling Frame Properties
Creating Control Blocks
Deleting Data Blocks

Working with Input Items

Creating Text Items
Controlling the Behavior and Appearance of Text Items
Creating LOVs
Defining Editors
Creating Check Boxes
Creating List Items
Creating Radio Groups

Working with Non Input Items

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

Working with Windows and Canvases

Overview of Windows and Canvases
Displaying a Forms Module in Multiple Windows
Creating a New Window

Displaying a Forms Module on Multiple Layouts

Creating a New Content Canvas

Creating a New Stacked Canvas

Creating a New Toolbar Canvas

Creating a New Tab Canvas

Producing Triggers

Grouping Triggers into Categories

Defining Trigger Components: Type, Code, and

Scope Specifying Execution Hierarchy

Using the PL/SQL Editor

Writing Trigger Code

Using Variables and Built-ins

Using the When-Button-Pressed and When-Window-Closed Triggers

Debugging Triggers

The Debugging Process

The Debug Console

Setting Breakpoints

Debugging Tips

Running a Form in Debug Mode

Stepping through Code

Adding Functionality to Items

Coding Item Interaction Triggers

Defining Functionality for Check Boxes

Changing List Items at Run Time

Displaying LOVs from Buttons

Populating Image Items

Populating and Displaying Hierarchical Trees

Interacting with JavaBeans

Run-Time Messages and Alerts

Built-ins and Handling Errors

Controlling System Messages

The FORM_TRIGGER_FAILURE Exception

Using Triggers to Intercept System Messages

Creating and Controlling Alerts

Handling Server Errors

Query Triggers

SELECT Statements Issued During Query Processing

WHERE and ORDER BY Clauses and the 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

Validation

Validation Process

Controlling Validation Using Properties

- Controlling Validation Using Triggers
- Performing Client-Side Validation with PJC's
- Tracking Validation Status
- Using Built-ins to Control When Validation Occurs

Navigation

- Navigation Overview
- Understanding Internal Navigation
- Using Object Properties to Control Navigation
- Writing Navigation Triggers: When-New--Instance, Pre- and Post- Triggers
- The Navigation Trap
- Using Navigation Built-ins in Triggers

Transaction Processing

- The Commit Sequence of Events
- Characteristics and Common Uses of Commit Triggers
- Testing the Results of Trigger DML
- DML Statements Issued During Commit Processing
- Overriding Default Transaction Processing
- Running Against Data Sources Other Than Oracle
- 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
- Referencing Objects by Internal ID
- Referencing Items Indirectly

Sharing 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 Forms Module
- Defining Multiple Form Functionality
- Sharing Data Among Modules