



Adobe® Flex® 3: Developing Rich Client Applications

Course Duration

3 days

Course Description

In this course, you will be introduced to Flex 3 and all its components through 13 modules and three lab exercises, including handling events, laying out an application using constraint-based layouts, using ActionScript Data Models, and more. This course is offered in an instructor-led, group-paced, classroom-delivery learning model with structured hands-on activities.

Course Prerequisites

To gain the most from this class, you should already:

- Be familiar with an object-oriented programming language, such as Java or C++
- Be familiar with XML terminology and use

Hardware Requirements*

You will need:

- Intel® Pentium® 4 processor
- 1 GB of RAM (2GB recommended)
- 500 MB of available hard-disk space (additional 500 MB required for plug-in configuration)

Software Requirements*

You will need:

- Microsoft® Windows® XP with Service Pack 2 or Windows Vista® Home (Premium or Basic), Business, or Ultimate
- Java™ Virtual Machine: Sun™ JRE 1.4.2, Sun JRE 1.5 (included), IBM® JRE 1.5, or Sun JRE 1.6
- Eclipse™ 3.2.2, 3.3 and 3.4 for plug-in configuration (Eclipse 3.3 recommended for Windows Vista)
- Adobe® Flash® Player 9

Performance-Based Objectives

- Explain the different technologies in the Adobe Flex 3 product line
- Understand the application flow of a Flex application and how it is transformed into a SWF file and delivered to the client

*Hardware and software requirements are only given here for Windows. For the MAC OS please check with the instructor



SOLUTION PARTNER
Gold

COURSE CONTENT

Lesson 1: Introducing Flex 3: Developing Rich Client Applications

- Introducing mastery learning
- Understanding the course format
- Reviewing the course prerequisites
- Reviewing the course outline

Lesson 2: Introducing Flex 3

- Understanding rich Internet applications
- Understanding Adobe Flex
- Understanding Adobe Flex Builder
- Understanding the foundation of the RIA
- Understanding the Flex application process flow
- Accessing remote data resources from a Flex application
- Getting help and other resources

Lesson 3: Getting Started with Flex Builder

- Understanding the relationship between Eclipse and Flex Builder
- Exploring the Flex Builder interface
- Creating a project and your first application
- Walkthrough 1: Creating a main application file and running it
- Debugging a Flex application
- Walkthrough 2: Debugging and using the debugging perspective
- Accessing help in Adobe Flex Builder

Lesson 4: Learning Flex Fundamentals

- Creating a simple Flex application
- Walkthrough 1: Creating your first Flex application
- Displaying images
- Walkthrough 2: Adding images to an application
- Laying out a Flex application with containers
- Walkthrough 3: Using basic layouts
- Using the Panel container
- Walkthrough 4: Separating application modules into panels
- Using the ControlBar container
- Walkthrough 5: Using a ControlBar container
- Adding user interface controls
- Creating data bindings between components
- Walkthrough 6: Creating data bindings
- Architecting an application with MXML components
- Walkthrough 7: Creating and instantiating a custom component
- Creating properties and methods for MXML components

Walkthrough 8: Creating properties and methods in a component

Lesson 5: Handling Events

- Understanding events
- Creating event handlers using inline ActionScript
- Walkthrough 1: Using inline Action Script
- Handling events within ActionScript functions
- Walkthrough 2: Using a function for an event handler
- Placing ActionScript functions in external files
- Understanding the Event object
- Walkthrough 3: Understanding the event object
- Using the addEventListener() method
- Walkthrough 4: Using the addEventListener() method

Lesson 6: Laying Out an Application using Constraint-Based Layout

- Understanding absolute positioning
- Positioning components within a Canvas container
- Walkthrough 1: Positioning components inside a Canvas
- Creating a constraint-based layout using Flex Builder
- Walkthrough 2: Using Design mode to implement a constraint-based layer
- Creating a constraint-based layer via MXML
- Walkthrough 3: Implementing a constraint-based layout in MXML
- Using Enhanced Constraints
- Using constraint-based layouts within nested containers
- Walkthrough 4: Using custom components in a constraint-based layout

Lab 1

Lesson 7: Using View States for Application Design

- Understanding view states
- Walkthrough 1: Creating two states for the Contact page
- Controlling view states
- Walkthrough 2: Switching states in the Contact form
- Reviewing the generated MXML code
- Walkthrough 3: Implementing view states using MXML
- Creating view states that include custom component states
- Walkthrough 4: Using custom components with their own view states

Lesson 8: Customizing the Application

Customizing Flex application look and feel

Modifying Styles to change look and feel

Using Themes

Walkthrough 1: Changing the look and feel of the Cafe Townsend application

Applying behaviors to components

Walkthrough 2: Add behaviors to the Cafe Townsend seating chart

Applying transitions to view state changes

Walkthrough 3: Resize the contact Cafe Townsend panel during the view states transition

Lab 2

Lesson 9: Using ActionScript Data Models

Using the MVC design pattern

Creating an MXML data model

Using ActionScript classes as a data model

Walkthrough 1: Instantiating an Object in MXML from an ActionScript class

Creating an ActionScript constructor with parameters

Walkthrough 2: Instantiating an Object in ActionScript from an ActionScript class

Defining class methods

Walkthrough 3: Adding Methods to an ActionScript class

Lesson 10: Exchanging Data Between Components Using Custom Events

Understanding the problem with bindings

Creating custom events

Walkthrough 1: Creating, dispatching, and handling a custom event

Sending data with a custom event

Walkthrough 2: Creating a custom event object and dispatching it

Lesson 11: Creating Data Entry Forms

Using the Form container

Walkthrough 1: Creating a data entry form component

Sharing form data

Walkthrough 2: Sharing form data with the application

Validating form data

Triggering validation with events

Walkthrough 3: Validating numeric input

Triggering validation with ActionScript

Walkthrough 4: Triggering validation with ActionScript

Lesson 12: Retrieving XML Data with HTTPService

Retrieving XML data at runtime

Handling results

Walkthrough 1: Retrieving data at runtime with HTTPService

Handling results using an event handler

Walkthrough 2: Using a result event

Handling faults

Walkthrough 3: Handling a fault event

Making HTTP requests to different domains

Walkthrough 4: Testing cross-domain policy

Making HTTP requests with parameters

Walkthrough 5: Using HTTPService with parameters

Lesson 13: Displaying Data Using the DataGrid

Using the DataGrid control

Specifying DataGrid columns

Walkthrough 1: Specifying DataGrid columns

Formatting DataGrid columns

Walkthrough 2: Formatting data in DataGrid columns

Using item renderers and item editors

Walkthrough 3: Using a drop-in item editor

Inline item renderers and item editors

Walkthrough 4: Using an inline item editor

Item renderer and item editor components

Walkthrough 5: Using a component item renderer

Using the TileList and HorizontalList

Walkthrough 6: Displaying data with a TileList and HorizontalList

Using events and selected items with list-based components

Walkthrough 7: Using a change event on the TileList

Lab 3



Lesson 18: Lab 3

Corporate Headquarters

Four Point Solutions Ltd.
106 Colonnade Road, Suite 210
Ottawa, Ontario, Canada
K2E 7L6
T. +1 613.907.6400
F. +1 613.225.1571

www.4Point.com

EMEA Headquarters

Four Point Solutions EMEA Ltd.
1 Exchange Place, IFSC
Dublin 1, Ireland
T. +44 203.0517.004
F. +35 387.814.6265

