

Adobe Flex 2.0 Developer Certification Examination Specification

Introduction

This is an exam specification for Adobe Flex Developer. The specification defines the content of the examination and the relevant background and experience suggested to become a certified Adobe Flex 2.0 Developer.

Exam Purpose

This examination is based upon the most critical job activities an Adobe Flex Developer performs. The skills and knowledge certified by this examination represent a professional level of expertise for a certified individual. The critical job activities of a Flex Developer are:

- Recognize effective use and reuse of code in Adobe Flex programming for application development, exception handling, troubleshooting, forms, and work with complex data types.
- Recognize key concepts in database interactions, client state management, and data exchange.

Background and Experience

The test specification is intended to address the knowledge and skill areas that demonstrate competency as an Adobe Flex Developer.

The knowledge level required is defined as having experience with non-code based software that uses visual objects in development and the following general suggested prerequisites:

- Two or more years experience with one or more programming languages.
- Experience in the applied use of an enterprise-level database server.
- One or more years of experience creating applications using Adobe Flex.

The Adobe instructor-led training courses that will assist in exam preparation are:

- Adobe Flex 2: Developing Rich Internet Applications
- Adobe Flex 2: Programming the Visual Experience
- Adobe Flex 2: Creating Dashboard Applications
- Using ActionScript 3 (final title TBD)
- Advanced Adobe Flex Data Services 2 (final title TBD)

The skills and knowledge measured by this examination are derived from an understanding of the jobs of current Adobe Flex Developers. A team of highly-qualified Adobe Flex developer experts defined the test content and wrote the test items.

Adobe Flex 2.0 Developer

Note: This examination blueprint includes weighting, test objectives, and example content. Example topics and concepts are included to clarify the test objectives; they should not be construed as a comprehensive listing of the content of this examination.

Exam Structure

The knowledge domains measured by this examination and the extent to which they are represented in the examination are shown in the table below.

Domain	% of Examination
1.0 Flex Application User Interface (UI) Creation	25%
2.0 Flex System Architecture and Design	20%
3.0 Flex Application Programming Fundamentals	35%
4.0 Interacting with Remote Data and Flex Applications	20%
Total	100%

Response Limits

The examinee selects, from four (4) or more response options, the option(s) that best completes the statement or answers the question. Distracters or wrong answers are response options that examinees with incomplete knowledge or skill would likely choose, but are generally plausible responses fitting into the content area defined by the test objective.

Test item formats used in this examination are:

Multiple-choice: The examinee selects one option that best answers the question or completes a statement.

Multiple-response: The examinee selects more than one option that best answers the question or completes a statement. The question will state how many options are correct; i.e., (Choose TWO).

Sample Directions: Read the statement or question and from the response options, select only the option(s) that represent the most correct or best answer(s) given the information provided.

Adobe Flex 2.0 Developer

1.0 Flex Application User Interface (UI) Creation

1.1 Identify and describe the basic UI controls used in a Flex application.

- Label, Text, TextInput, Button, DataGrid, etc.

Enabling Sub-skills:

- Identify how to use the following with controls: labelFunction, item editor, dataProvider, and itemRenderers.
- Identify how to display data by using a DataGrid and DataGridColumns.
- Identify how to display data by using a list-based controls (i.e., DataGrid, List, ComboBox, HorizontalList, TileList).

1.2 Identify the purpose of UI containers and when to use them.

Enabling Sub-skills:

- Recognize basic containers and their use (Application, HBox, VBox, Canvas).
- Recognize navigator containers and their use (TabNavigator, Accordion, ViewStack, TabBar, LinkBar).
- Recognize layout containers and their use (Panel, ControlBar, Form, ApplicationControlBar, Tile).
- Recognize constraint-based layout containers and their use.

1.3 Identify how to change the look and feel of a design by using API styles and style sheets.

1.4 Recognize how to customize changes, control-related presentation layouts and navigation in an application by using view states, and using transitions and effects.

Adobe Flex 2.0 Developer

2.0 Flex System Architecture and Design

- 2.1 Recognize how UML and Use Cases are used in object-oriented design.
- 2.2 Identify the relationship of design patterns with respect to system architecture.
 - Model, View, Controller, Singleton, Assembler, Data Access Object (DAO), and Value Object/Transfer Object/Bean.
- 2.3 Recognize and describe event-based programming model in events.
 - Event bubbling, custom event classes, and event listening
- 2.4 Identify how to build loosely-coupled components.
- 2.5 Recognize how Inheritance, Encapsulation, and Polymorphism works.
- 2.6 Recognize the structure of a J2EE architecture.

Enabling Sub-skills:

- Recognize how and why to use **.war** and **.jar** files.
- Identify and describe the directory structure of a J2EE application.
- Recognize the characteristics of a Java class used by a Flex application.
- Given a Java interface, describe that interface. Interfaces include: JMS, JTS, JNDI, Hibernate, and JDBC.
- Describe the use of servlets and Java Server Pages.

Adobe Flex 2.0 Developer

3.0 Flex Application Programming Fundamentals

- 3.1 Identify the properties and relationships between classes and objects and how they work in Flex.

Enabling Sub-skills:

- Identify class types; abstract, dynamic, final, inner.
- Recognize the difference between a class and an interface.
- Recognize the purpose of access modifier and when to use that modifier (public, private, protected, internal, static).
- Identify the purpose and how to use properties of classes and objects.
- Identify the differences between variables, properties, and constants.
- Identify the purpose and how to use methods.
- Recognize how bindings are used to access values of objects.
- Format data by using built-in and custom formatter classes. Formatter classes include: NumberFormatter, CurrencyFormatter, DateFormatter.

- 3.2 Identify how to display data in Flex.

Enabling Sub-skills:

- Recognize how to use the IViewCursor interface to locate, add, remove, and bookmark data in a collection.

- 3.3 Identify how to manipulate data in Flex.

Enabling Sub-skills:

- Store complex data by using Collection classes. Collection classes include: ArrayCollection, XMLListCollection.
- Filter data by using a filter function.
- Sort data by using the Sort class and associated SortField objects.

- 3.4 Recognize how to validate data by using built-in and custom validator classes. (NumberValidator, CurrencyValidator, CreditCardValidator, DateValidator)

Enabling Sub-skills:

- Identify how to create a custom validator by using regular expressions.

- 3.5 Recognize how to create and use ActionScript classes.

Enabling Sub-skills:

- Identify how to use the package and class keywords to create an ActionScript class.
- Recognize how to use a constructor in an ActionScript class.
- Identify how to create properties and methods in an ActionScript class.

- 3.6 Identify the purpose and uses of handling events in Flex.

Enabling Sub-skills:

- Identify the purpose of user events; i.e., click, change, mouseOver.
- Identify the purpose of system events; i.e., creationComplete, initialize, show.

Adobe Flex 2.0 Developer

- Recognize how to create an event handler (inline, in an ActionScript function, in an external file).
- Identify the purpose of custom events.
- Recognize how to create a custom event class.
- Identify how to declare, dispatch, and handle a custom event.

3.7 Recognize how to use XML data in Flex.

Enabling Sub-skills:

- Recognize how to access attributes and elements of XML documents by using E4X operators (e.g., dot, parentheses, attribute, and descendant accessor).
- Recognize how to populate a Tree control with XML data.
- Identify how to store XML data by using XML data structures (XMLList, XMLListCollection).

3.8 Recognize how to create and use custom components in Flex.

Enabling Sub-skills:

- Identify how to use and when to use custom components.
- Identify the advantages of custom components.
- Recognize how to create a custom component by using a control as a base tag and using a container as a base tag.
- Identify how to create properties and methods in a custom component.
- Recognize how to Instantiate a custom component.
- Recognize how to use the properties and methods of an instantiated custom component.
- Recognize how to dispatch data from a custom component by using a custom event.

3.9 Recognize how to display data using different chart types, formats, and legends.

Enabling Sub-skills:

- Recognize data structures for charting.
- Identify chart series.
- Identify how to create and use chart legends.
- Identify how to format chart axes.
- Identify how to format data tips.
- Recognize chart specific formatting; i.e., Line/Area, Pie/Doughnut.
- Identify how to create charts with multiple data series.
- Recognize ranges selection for axes.
- Recognize adding/removing series dynamically.
- Identify how to use graphical renderers.
- Identify how to style chart components.
- Identify how to style chart gridlines.
- Identify how to style chart axes.

Adobe Flex 2.0 Developer

4.0 Interacting with Remote Data and Flex Applications

4.1 Recognize how to use Remote Procedure Call (RPC) services.

Enabling Sub-skills:

- Identify how to retrieve remote data by using the HTTPService.
- Identify how to retrieve remote data by using the WebService.
- Identify how to retrieve remote data by using the RemoteObject.

4.2 Recognize how to manage data using Flex Data Services (FDS).

Enabling Sub-skills:

- Recognize how to install and configure Flex Data Services.
- Recognize the functionality and features of Flex Data Services; i.e., Data Management Service, Message Service, and RPC services.
- Identify how to configure destinations, adapters, and channels.
- Recognize how to use the Data Management service to synchronize data between clients and servers.
- Identify how to publish, retrieve, and process messages.
- Recognize how to use FDS to implement occasionally connected clients.
- Recognize how to use FDS to handle paging of large datasets.