



Overview

Visualforce is the Force.com component-based user interface framework. It uses the MVC pattern, is extensible (you can create your own components), and supports the standard platform user interface, as well as a completely custom look and feel.

Getting Started

Enable *Developer Mode* by clicking [Setup | My Personal Information | Personal Information](#). Select the *Development Mode* checkbox to enable an inline Visualforce editor. To create a page, navigate to a URL based on your current instance, like:

`https://<instance>.salesforce.com/apex/helloWorld`

All Visualforce page markup lies within an `<apex:page>` tag. Here's a simple page:

```
<apex:page showHeader="false">
  <h1>Hello World</h1>
</apex:page>
```

All tags take an optional or obligatory set of attributes, such as `showHeader` above. Use the inline editor auto-completion, or the *Component Reference*, to determine the attributes for components.

List all Visualforce pages and Apex classes by navigating to [Setup | Develop](#).

Example

Visualforce supports auto-generated *standard controllers*, as well as *extensions*, to minimize the amount of coding needed. Here's an example of a Visualforce page that uses a *custom controller*.

```
<apex:page showHeader="false" controller="Hello">
  <apex:form>
    <apex:inputText value="{!theText}"/>
    <apex:commandButton value="Go"
      action="{!action}" reRender="dynamic"/>
  </apex:form>
  <apex:outputPanel id="dynamic">
    {!theText}
  </apex:outputPanel>
</apex:page>
```

This Visualforce page displays an input field and a button labeled **Go**. When clicked, it sends the value of the field to the controller, which performs an action on it – here, it duplicates the input. The page renders the result using an AJAX update. Here's the controller:

```
public class Hello {

  public PageReference action() {
    theText = theText + theText;
    return null;
  }

  public String theText {get;set;}
}
```

Core Tags

Here are core tags that often make up a Visualforce page:

<code>apex:page</code>	The obligatory outer tag
<code>apex:includeScript</code>	A link to a JavaScript library, often in a <i>static resource</i>
<code>apex:stylesheet</code>	A link to a CSS stylesheet, often in a <i>static resource</i>

Template System

Template components let you create Visualforce pages that act as templates, having named areas – *inserts* – that must be filled when the template is used.

<code>apex:insert</code>	Declares a named area that must be defined
<code>apex:composition</code>	Adds the defined template
<code>apex:define</code>	Provides content for the inserts

AJAX and JavaScript

The *reRender* attribute on action links such as *commandButton* let you refresh a part of a page. The following tags provide additional AJAX and JavaScript support:

<code>apex:actionFunction</code>	Define JavaScript functions to be called from JavaScript code
<code>apex:actionPoller</code>	A timer that sends an AJAX update request at specified interval
<code>apex:actionRegion</code>	Defines an area in a page that demarcates which components should be processed when AJAX requests generate
<code>apex:actionStatus</code>	A component that displays the status of an AJAX update request
<code>apex:actionSupport</code>	Adds asynchronous refresh support to a component

Core Form

Here are the core components used in forms. These should be embedded within a single *form* component on a page:

<code>apex:form</code>	Container for input components
<code>apex:commandButton</code>	A form button used to submit/reset a form
<code>apex:commandLink</code>	An HTML link that performs an action
<code>apex:inputCheckbox</code>	An HTML input element of type checkbox
<code>apex:inputField</code>	Input element that corresponds to a field on a custom object
<code>apex:inputFile</code>	An input field to upload a file
<code>apex:inputHidden</code>	An HTML input element of type hidden
<code>apex:inputSecret</code>	An HTML input element of type password
<code>apex:inputText</code>	An HTML input element of type text
<code>apex:inputTextArea</code>	A text area input element
<code>apex:selectList</code>	A list of options for radio/checkboxes
<code>apex:selectRadio</code>	A set of related radio button input elements, displayed in a table
<code>apex:selectOption</code>	A possible value for a selectCheckboxes or selectList component
<code>apex:selectCheckboxes</code>	A set of related checkbox input elements, displayed in a table
<code>apex:selectOptions</code>	A collection of possible values for the selectCheckboxes, selectRadio, or selectList components

Core Output

Mirroring many of the Core Form Components, these components are used for outputting text:

<code>apex:outputField</code>	A read-only display of a label and value for fields on custom objects
<code>apex:outputLabel</code>	A label for an input or output field
<code>apex:outputLink</code>	A link to a URL
<code>apex:outputPanel</code>	A set of content that is grouped together, rendered with an HTML span tag, div tag, or neither
<code>apex:outputText</code>	Displays styled or escaped text for a page

Tables and Lists Output

These components are used to generate tables or lists by iterating over a collection:

<code>apex:dataTable</code>	An HTML table
<code>apex:column</code>	A single column in a table
<code>apex:dataList</code>	An ordered or unordered list of values
<code>apex:facet</code>	A placeholder for content that is rendered in a specific part of the parent component
<code>apex:panelGrid</code>	An HTML table element that lays out its components in consecutive cells
<code>apex:panelGroup</code>	A container for multiple child components so that they can be displayed in a single panelGrid cell

Miscellaneous HTML

These components generate HTML for embedding flash, iframes and images:

apex:flash	A Flash movie, rendered with object and embed tags
apex:iframe	Creates an iframe in a page
apex:image	A graphical image, rendered with an img tag

Miscellaneous Visualforce

These components provide Visualforce messages, iteration, and include functionality:

apex:repeat	An unformatted iteration component
apex:message	A message for a specific component, such as a warning or error
apex:messages	All messages that were generated for all components on the current page
apex:include	Inserts a second Visualforce page into the current page
apex:param	Lets parameters be fed into other components
apex:scontrol	An inline frame that displays an s-control
apex:variable	A local variable representing an expression

Default Look and Feel

CRUD and other screens have a standard look and feel. The following components generate output that conforms to that look and feel:

apex:detail	The standard detail page for a particular object
apex:enhancedList	The list view picklist for an object, including its associated list of records
apex:listViews	The list view picklist for an object, including its associated list of records
apex:relatedList	A list of records that are related to a parent record with a lookup or master-detail relationship
apex:pageBlock	Defines a visual block within a page
apex:pageBlockButtons	Defines buttons related to a block
apex:pageBlockSection	A section of data within a pageBlock component
apex:pageBlockSectionItem	A single piece of data in a pageBlockSection that takes up one column in one row
apex:pageBlockTable	A list of data displayed as a table within either a apex:pageBlock or pageBlockSection component
apex:pageMessage	Displays custom messages
apex:pageMessages	Displays all messages that were generated for all components on the page
apex:panelBar	A page area that includes one or more panelBarItems that can expand
apex:panelBarItem	A section of a panelBar that can expand or retract
apex:sectionHeader	A title bar for a page
apex:tab	A single tab in a tabPanel
apex:tabPanel	A page area that displays as a set of tabs
apex:toolbar	A toolbar that can contain any number of child components
apex:toolbarGroup	A group of components within a toolbar

Custom Components

Create your own components that will reside in the c namespace, for example `<c:helloWorld/>`:

apex:component	The outer tag that starts the definition of a component
apex:attribute	A definition of a typed attribute on a custom component
apex:componentBody	A place holder for content injected into a component

Sites

Sites is based on Visualforce pages, and the following tag provides additional functionality:

site:googleAnalyticsTracking	Used to integrate Google Analytics
------------------------------	------------------------------------

Ideas

Applications that integrate Ideas have additional components available:

ideas:detailOutputLink	A link to the page displaying an idea
ideas:listOutputLink	A link to the page displaying a list of ideas
ideas:profileListOutputLink	A link to the page displaying a user's profile
apex:vote	Displays the vote control for an idea

Messaging

Visualforce can also be used to create email templates:

messaging:attachment	Appends an attachment to the email
messaging:emailHeader	Adds a custom header to the email
messaging:emailTemplate	Defines a Visualforce email template (*)
messaging:htmlEmailBody	The HTML version of the email body
messaging:plainTextEmailBody	The plain text (non-HTML) version of the email body

Static Resources

Upload static resources, such as stylesheets and images, into a zip, and then reference them using the URLFOR formula and \$Resource merge field. For example, if you've uploaded images/Blue.jpg into a zip called TestZip, reference it like this:

```
<apex:image url="{!URLFOR($Resource.TestZip, 'images/Blue.jpg') }"/>
```

Global Variables

These reference general information about the current user and your organization on a Visualforce page:

\$Action	Reference standard actions on objects
\$API	Refer to API URLs or sessions IDs
\$Component	Reference a component from within, for example, JavaScript
\$CurrentPage	Reference name, parameters, URL of current page
\$Label	References a custom label
\$ObjectType	Access metadata of objects
\$Organization	Access information about the company profile
\$Page	The way to reference a Visualforce page
\$Resource	Used for accessing static resources
\$SControl	References an existing custom s-control by name
\$Site	Reference site information, such as domain name
\$System.OriginDateTime	Represents the literal value of 1900-01-01 00:00:00
\$User	Information about the current user
\$UserRole	The current user's role information
\$Profile	The current user's profile

Apex Support

Here are Apex classes, available in the ApexPages namespace, which can be used to reference Visualforce functionality:

Action	Create an invoke actions on an object
Message	Create messages to be displayed, usually for errors
PageReference	A reference to a Visualforce page
SelectOption	Specifies one of the possible values for the selectCheckboxes, selectList, or selectRadio components
StandardController	Reference the standard Visualforce controllers
StandardSetController	Reference standard set controllers for iteration