
OurUniversity
EFMD CLIP ACCREDITED
Force.com Visualforce Pages
salesforce
global strategic
consulting partner
Lesson 03 : Visualforce Pages

Lesson Objectives

By the end of this lesson, you will be able to:

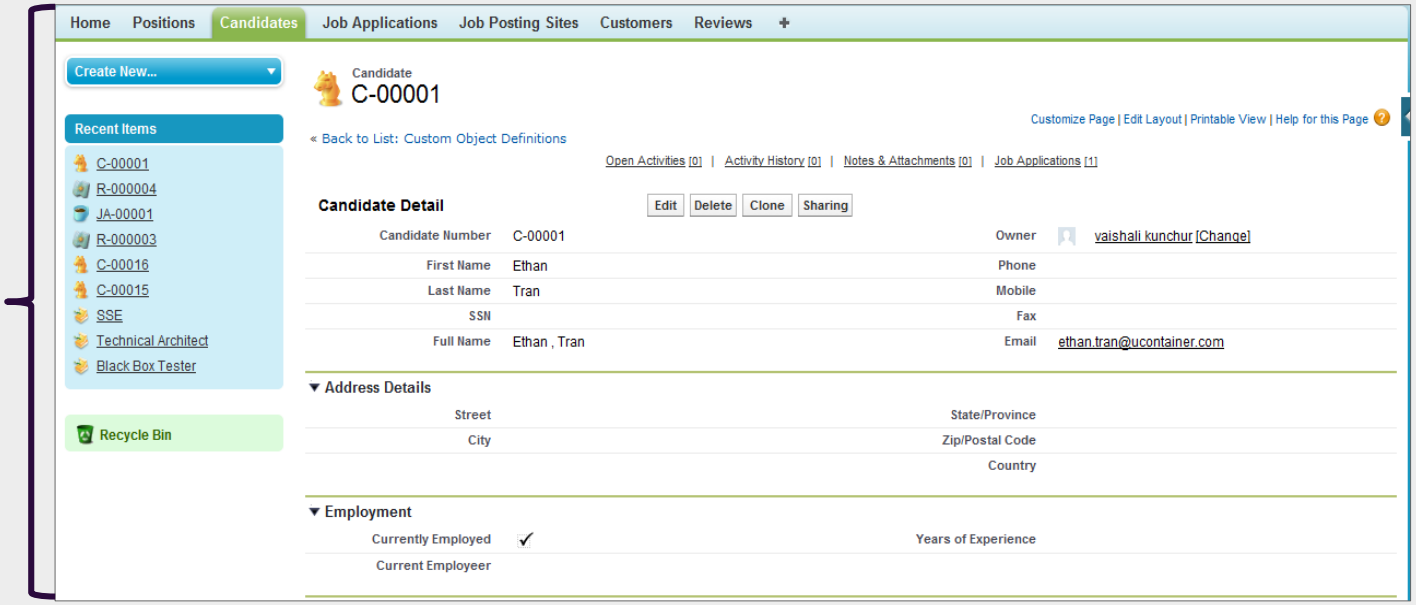
- List Key tags and define what their attributes control
- Create Visualforce pages that use these tags to create page layouts

Core Components
Layout Components

- This group of components provides overall structure to the page or provides templates or frames to insert other content
- For the most part, they do not bind directly to Salesforce data, but are focused more on containers and areas to put those data bound components

Core Components
<apex:page>

<apex:page>



Core Components
<apex:page>

This represents a single Visualforce page.
All pages must be surrounded by an <apex:page> tag set
- This include pages used within other page layout

Attributes include:

- **standardController or controller(String)**: Salesforce object used to control behavior. Standard controllers can also use the **extensions** attribute
- **title** (String): title displayed on the page
- **tabStyle** (String): Salesforce object used to copy the color and style for this page
- **sidebar** (Boolean): whether the sidebar should be used (default = true)
- **action** (Action method): invokes an action method on the controller to be run when the page is first requested
- **renderAs** (String): a supported content converter (currently only pdf)

Core Components
<apex:variable/>

This tag provides a local variable that can be used to replace an expression to reduce long and repetitive text.

Many other tags, especially iteration tags, also have variable attributes build in

Attributes include:

- **value** (Object): an expression that can be represented by the new variable
- **var** (String): the name of the new variable

Core Components
Static resources

Static resources are a new kind of Salesforce storage, designated for use with Visualforce.

Static resources are items required by your Visualforce pages, such as archives, images, style sheets, Javascript functions, PDFs,etc.

- These resources can be referenced using the $Resource global variable
- They can be a collection of related files in a directory hierarchy (.zip, .jar)
- This is the recommended method over uploading these files to the Document tab, as these resources can be cached for better performance

Upload files via Your Name | Setup | develop | Static Resources

- There is a 5 MB limit per file and a 250 MB overall limit for static resources

Core Components
Static Resources Components

The following tags can all be used to include other HTML – renderable technologies into your page.

- <apex:flash>
- <apex:image>
- <apex:stylesheet>

These can be stored in the static Resources object or elsewhere on the internet.

Core Components
Redirect to Static Resources

You can also use the action attribute on the page component to redirect to a static resource, such as a custom help PDF file:
action = " {!URLFOR($Resource.filename)}">

1. <apex:page sidebar="false" showHeader="false
   standardStylesheets = "false"
   action="{!URLFOR($Resource.filename)}">
2. </apex:page>

Core Components
Style Sheets

Use the <apex:stylesheet> tag to add additional styles to a page

- All tags that produce HTML have pass-through **style** and **styleClass** attributes to allow you to use your style with any visual component

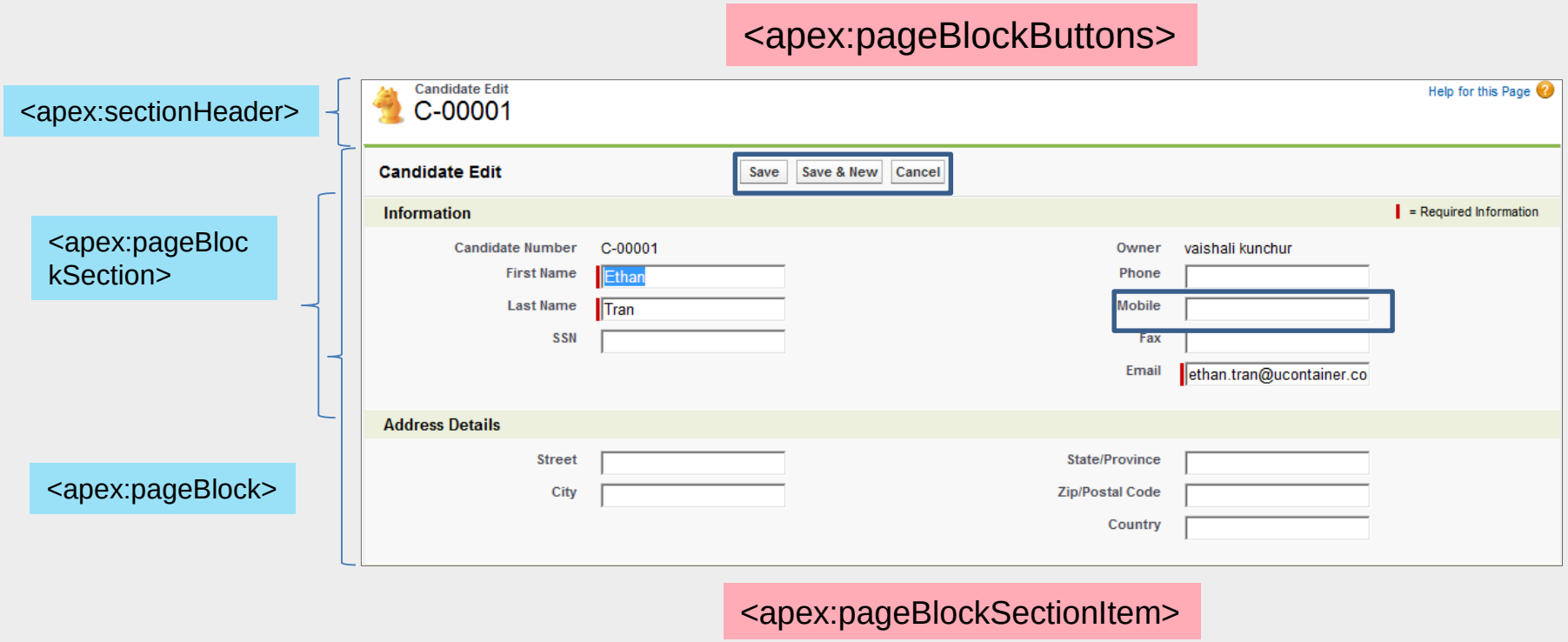
Salesforce style sheets are located in the /sCSS/ directory of your Salesforce instance

- **dStandard.css**: contains most style definitions for standard objects/tabs
- **/sCSS/12.0/Theme2/allCustom.css**: contains style definitions for custom objects/tabs

Note: Visualforce is not designated to be a way to fully re-brand a standard Salesforce application!

Default Look and Feel Components <apex:pageBlock>

There are a series of pageBlock tags that help build out pages and use the Salesforce style sheet by default



Default Look and Feel Components <apex:pageBlock>

This tag creates an area of a page that is similar to a detail page but with no default content.

Attributes include:

- **mode** (String): either **detail** or **edit** mode. **Edit** mode does not show field lines (detail)
- **title** (String): title displayed on the page block
- **tabStyle** (String): Salesforce object used to copy the color and style for this page block

The pageBlock **tags can include** facet **tags, which are child tags that are often used to add or override headers and footers to areas or tables.**

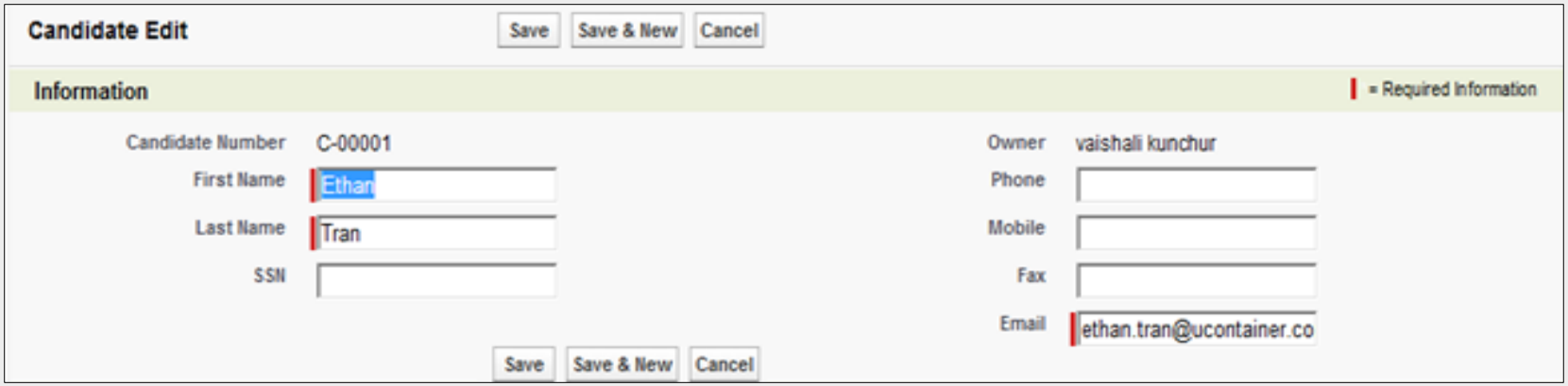
Default Look and Feel Components <apex:pageBlockButton>

This tag creates a set of buttons that are styles like standard Salesforce buttons

- The individual buttons are created using the commandButton tag

Attributes include:

- Location (String): either top, bottom, or both.



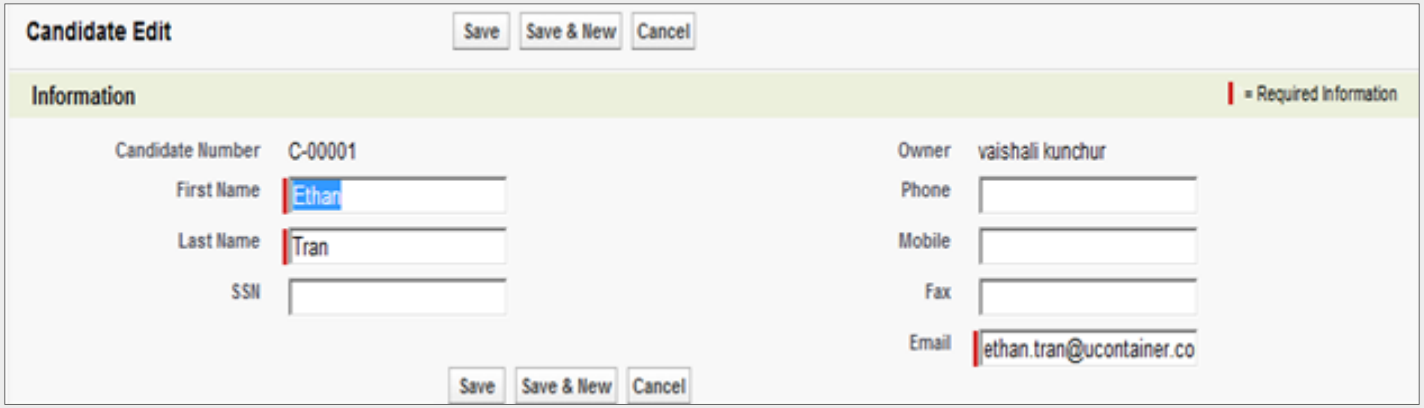
Default Look and Feel Components <apex:pageBlockSection>

This tag must be used within a pageBlock component to create a section with one or more columns.

- Each column has two cells: one for the label and one for the value.
- Add items to cells with the **inputField** or **outputField** components.
- Or you can use **pageBlockSectionItem** components for items not based directly on Salesforce object fields.

Attributes include:

- collapsible (Boolean): if the section can collapse. (true).
- Columns (Integer): number of columns, each with two cells. (2).

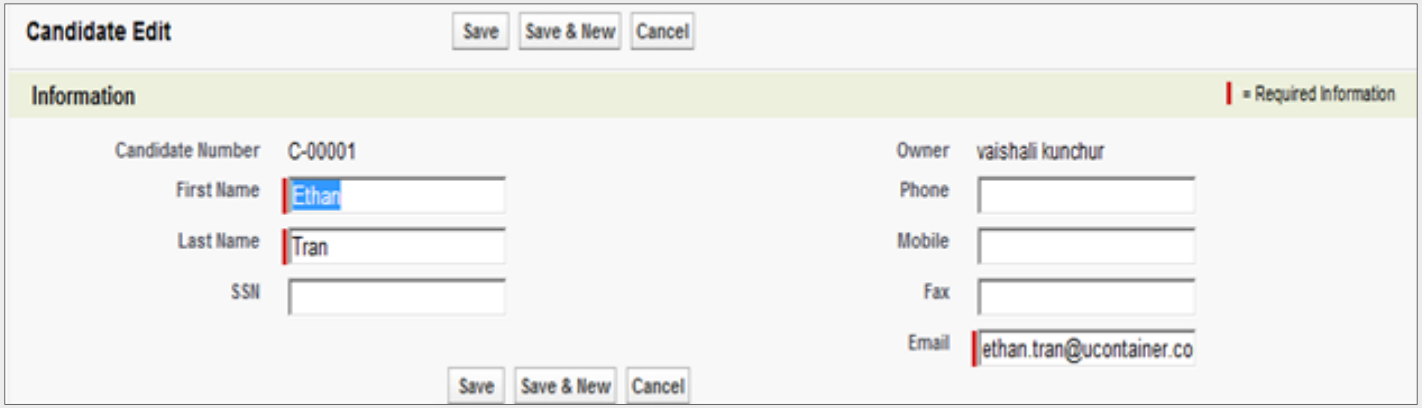


Default Look and Feel Components <apex:pageBlockSectionItem>

This tag must be used within a pageBlockSection component to create a pair of cells as an item in a column, instead of using inputField or other tags that automatically create name/value data pairs.

Each column has two cells: one for the label and one for the value:

- These can be added with **outputLabel** tags and **inputText** or other UI widget components.
- If you need to bundle more than one item per cell, wrap them in an **<apex:outputPanel>** components.

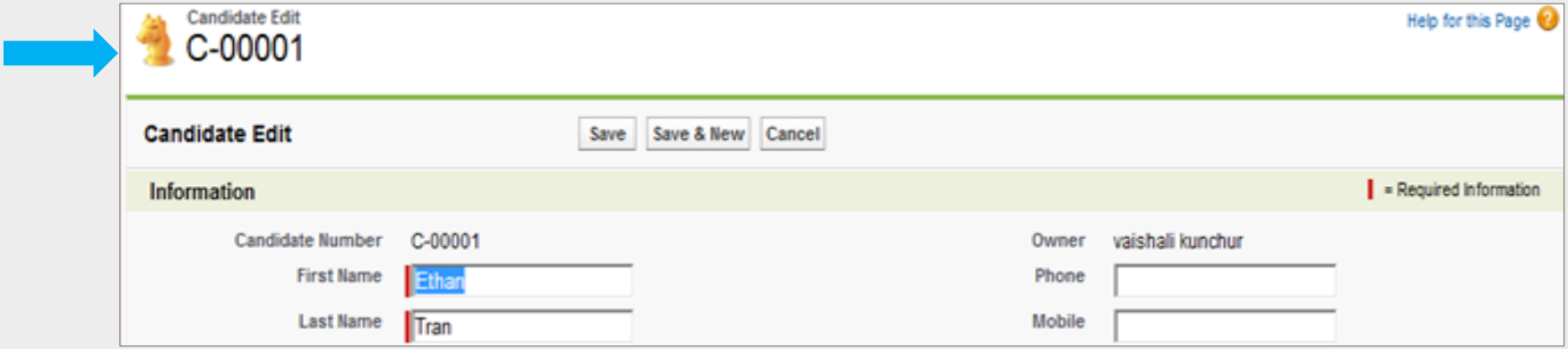


Default Look and Feel Components <apex:sectionHeader>

This tag creates the standard header bar displayed under the tabs in the Salesforce UI.

Attributes include:

- **title** (String): the text display at the top of the bar
- Subtitle (String): the text displayed under the title
- help (String): the URL for the page's help file. The link appears on the right



Default Look and Feel Components
Other UI Features

There are a variety of other tags that can be used to create standard Web features, such as:

<apex:toolbar> & <apex:toolbarGroup>

- Create a toolbar with clickable items
- Cluster related items together using groups

<apex:tabPanel> & <apex:tab>

- Create a panel of tabs with different content
- Functions similarly to tabs in Salesforce, but content for all tabs is loaded when the page is loaded

Default Look and Feel Components
Other UI Features

<apex:panelBar> & <apex:toolBarItem>

- Create panels that can expand and collapse when a user clicks the solid panel bar
- Opening another panel bar item closes any items that were previously open

<apex:panelGrid> & <apex:panelGroup>

- Place items into an HTML grid
- Use groups to place multiple items in the same cell
- Does not process a set of data with an iteration variable

Demo

Creating a Visualforce Detail Page

Default Look and Feel Components
Coarse Metadata Components

These tags provide a large amount of generated code for surprisingly little work to create familier Salesforce structures

- detail
- relatedList
- listView
- enhancedList

They do not allow for much customization to the generated areas

- Although, it is possible to build up identical sections using finer metadata and layout components

Default Look and Feel Components <apex:detail/>

This single tag creates the standard detail page layout for this object.
Attributes include:

- subject (String): the ID of the record that should provide the data for the detail component
- relatedList (Boolean): if the standard related lists should be included on the page (true)
- relatedListHover (Boolean): if the standard hover links are included (true)

Default Look and Feel Components <apex:relatedList/>

This single tag creates just the related list for records related to a parent record

Attributes include:

- list (String): the name of the child relationship from which to return the records
- Subject (String): the ID of the record that should provide the data for the related list
- pageSize (Integer): the number of records to display by default (5)

Default Look and Feel Components <apex:listView/>

This single tag creates the list view picklist for an object, usually displayed on the main tab for an object.

Attributes include:

- type (String): the object for which list views are displayed

Default Look and Feel Components <apex:enhancedList>

The enhancedList is similar to listViews, but provides more flexibility by providing additional components, such as:

- customizable (by the current user)
- Height
- rowsPerPage
- width

Default Look and Feel Components <apex:repeat>

This tag simply iterates over the data of a collection for a structure that you specify.

- It is typically usedused if one of the other iteration components does not meet the requirements.

Attributes include:

- **value** (Object): reference the collection of data
- **rows** (Integer): the number of items in the collection that are rendered. (0=all rows)

Default Look and Feel Components
Message Components

The <apex:message/> tag passes warning or error messages for specific component into the Visualforce page

The plural version, <**apex:message/**>, is for all messages generated for all components on the current page

There are also **pagemessage** and **pagemessages** are tags that use the standard Salesforce style

Use this tag to display messages, such as validation errors, to users that are usually available through the standard Salesforce UI

Attributes include:

for (String): ID of the component with which the label should be associated

title (String): the text displayed next to the output link

style (String): the is used to display the component

tooltip (Boolean): specifies whether the detail portion of the message is displayed as a tooltip

Summary

Layout Components
Static Resources Components
Coarse Metadata Components
Other UI Features
Message Components

Module Review

How many controllers can a page have? Where is the controller for a page assigned?

There are a series of layout components that all help re-create the traditional Salesforce page layout style very easily. What name do they share?

What are the names of the coarse metadata tags?