



Getting Started with Salesforce CRM

Getting Started with SALESFORCE Part XV

Description:

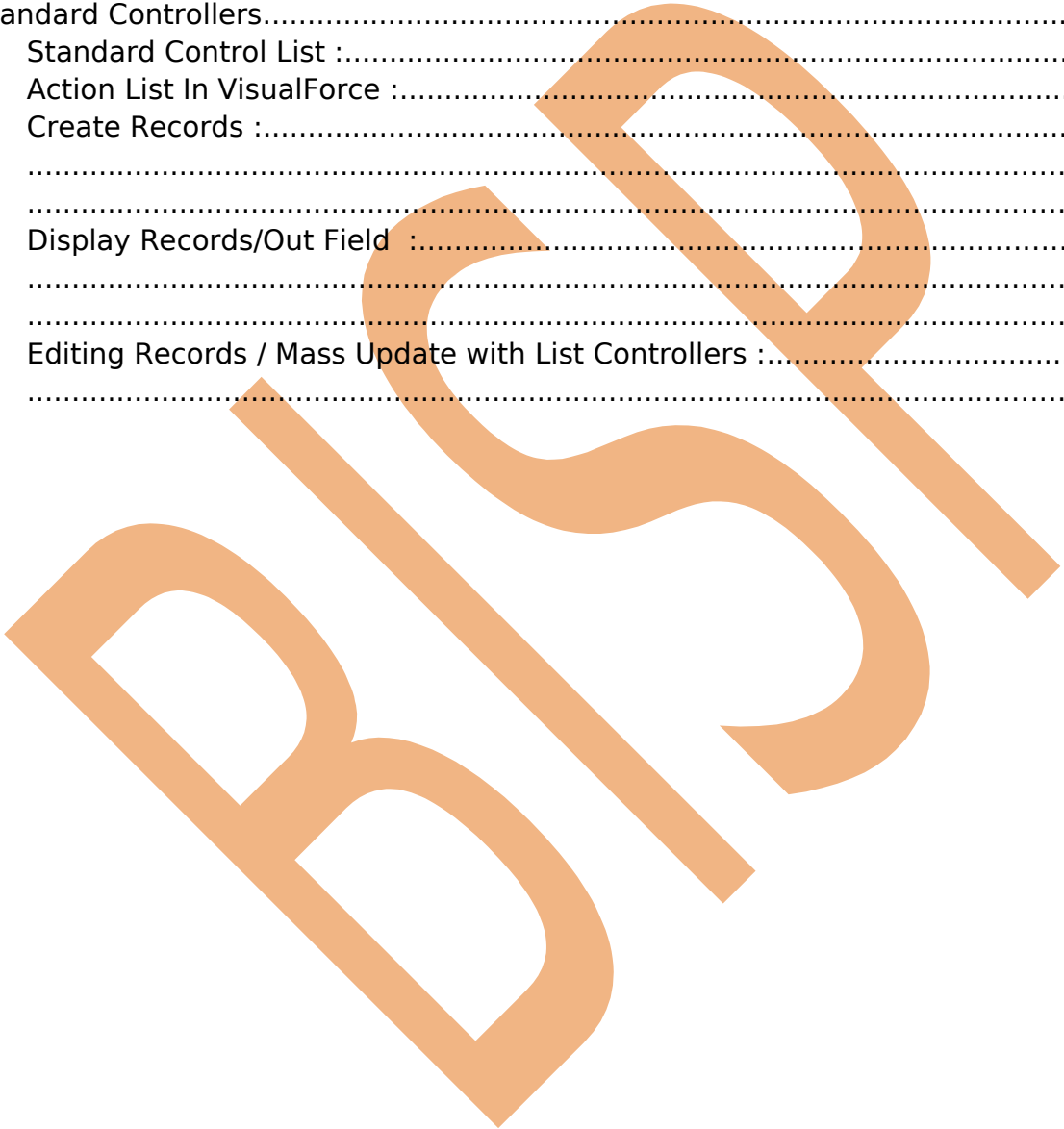
BISP is committed to provide BEST learning material to the beginners and advance learners. In the same series, we have prepared a complete end-to end Hands-on Beginner's Guide for Salesforce. The document focuses on VisualForce Page ,standard control and action list. **Join our professional training program and learn from experts.**

History:

Version	Description Change	Author	Publish Date
0.1	Initial Draft	Chandra Prakash Sharma	18 th Nov 2012
0.1	Review#1	Amit Sharma	18 th Nov 2012

Contents

Contents.....	2
What is VisualForce.....	3
Why use VisualForce :.....	4
How To Create VisualForce Page	4
Standard Controllers.....	7
Standard Control List :.....	8
Action List In VisualForce :.....	8
Create Records :.....	9
.....	10
.....	10
Display Records/Out Field :	10
.....	11
.....	11
Editing Records / Mass Update with List Controllers :.....	11
.....	12



What is VisualForce

VisualForce is framework that allows developers to build refined, custom user interfaces that can be hosted natively on the Force.com platform. The VisualForce framework includes a tag-based markup language, similar to HTML.

In VisualForce page you can use html code, JavaScript, css, Ajax, Flesh.

Developers can use VisualForce to create a VisualForce page definition. A page definition consists of two primary elements:

- VisualForce markup.
- A VisualForce controller.

VisualForce Markup :-

VisualForce markup consists of VisualForce tags, HTML, JavaScript, or any other Web-enabled code embedded within a single `<apex: page>` tag. The markup defines the user interface components that should be included on the page, and the way they should appear.

VisualForce Control :-

A VisualForce controller is a set of instructions that specify what happens when a user interacts with the components specified in associated VisualForce markup, such as when a user clicks a button or link.

The screenshot shows a Salesforce VisualForce page titled "Your Account Owners". The page features a table of account owners and a section for "Platinum and Gold SLA Customers". Red boxes and arrows highlight specific Apex tags used in the page:

- `<apex:page>` points to the top navigation bar.
- `<apex:pageBlock>` points to the main content area.
- `<apex:enhancedList>` points to the "Platinum and Gold SLA Customers" table.
- `<apex:dataTable>` points to the "Your Account Owners" table.

Account Name	Owner
Account	Admin User
Burlington Textiles Corp of America	John Franklin
Pyramid Construction Inc.	Admin User
Grand Hotels & Resorts Ltd	Admin User
Express Logistics and Transport	Admin User
University of Almania	Admin User
United Oil & Gas Corp.	Admin User
GenePoint	Admin User
United Oil & Gas, UK	Admin User
United Oil & Gas, Singapore	Admin User
Edge Communications	John Franklin

Action	Account Name	Account Site	Billing State/Province	Phone	Owner Alias
<input type="checkbox"/> Edit Del	Account	www.salesforce.com			Admin
<input type="checkbox"/> Edit Del	Burlington Textiles Corp of Am		NC	(336) 222-7000	fran
<input type="checkbox"/> Edit Del	Edge Communications		TX	(512) 757-6000	fran
<input type="checkbox"/> Edit Del	Express Logistics and Transport		OR	(503) 421-7800	Admin
<input type="checkbox"/> Edit Del	GenePoint		CA	(650) 867-3450	Admin
<input type="checkbox"/> Edit Del	Grand Hotels & Resorts Ltd		IL	(312) 596-1000	Admin
<input type="checkbox"/> Edit Del	Pyramid Construction Inc.			(014) 427-4427	Admin
<input type="checkbox"/> Edit Del	United Oil & Gas Corp		NV	(755) 649 8800	Admin

Why use VisualForce :

- To create custom interfaces.
- To build wizards.
- To define custom navigation patterns.

How To Create VisualForce Page

Click on **Setup** > **develop** > **Pages**
Then Click on New button.

The screenshot shows the Salesforce Developer Console interface. At the top, the user 'yogesh sharma' is logged in. The navigation menu includes 'Home', 'Positions', 'Job Applications', 'Departments', 'Employees', 'wiki', 'Employment websites', 'Employee List', 'Reports', 'Dashboards', and 'All Details'. The left sidebar shows the 'Build' menu with 'Develop' selected, and 'Pages' highlighted. The main area displays 'Visualforce Pages' with a table of existing pages. A 'New' button is highlighted in the top right of the table.

Action	Label	Name	Namespace Prefix	Api Version	Description	Created By Alias	Created Date	Last Modified By Alias	Last Modified Date
Edit Del Security Info	My New	My New		28.0		yshar	10/4/2013 11:34 AM	yshar	10/4/2013 11:34 AM

Step 1 :-

Visualforce Page



The screenshot shows the 'Page Edit' interface. The 'Page Information' tab is active, showing fields for 'Label' and 'Name' (both set to 'My First Page') and a 'Description' field containing 'This page is store basic information.'. Below this, there are checkboxes for 'Available for Salesforce mobile apps' and 'Require CSRF protection on GET requests'. The 'Visualforce Markup' tab is also visible, showing the following code:

```
1 <apex:page>
2 <!-- Begin Default Content REMOVE THIS -->
3 <h1>Congratulations</h1>
4 This me First VisualForce Page
5 <marquee> Welcome To BISP Solutions Ins. </marquee>
6 <!-- End Default Content REMOVE THIS -->
7 </apex:page>
```

A red box highlights the code in the 'Visualforce Markup' tab, and a red arrow points to it with the text: 'There you can write code this code simple write in html type'.

After Save this page. click on this icon for view Home page.

Click here for view this page

Action	Label	Name	Namespace Prefix	Api Version	Description	Created By Alias	Created Date	Last Modified By Alias	Last Modified Date
Edit Del Security 	Home	Home		29.0	This page is store basic information.	yshar	11/11/2013 2:55 PM	yshar	11/11/2013 2:55 PM
Edit Del Security 	My New	My New		28.0		yshar	10/4/2013 11:34 AM	yshar	10/4/2013 11:34 AM

If you have getting error for view page solve this problem following some steps you can see below.

Click on **Setup > Manage Users > Users** then click on Edit link button for edit user settings.

Action	Full Name	Alias	Username	Last Login	Role	Active	Profile	Manager
Edit	Chatter Expert	Chatter	chattv.00d900000000000000x9eam.ih6mvzuphxt@chatter.salesforce.com			<input checked="" type="checkbox"/>	Chatter Free User	
Edit	sharma_vogesh	yshar	vogesh@lvs.com	11/12/2013 8:10 AM		<input checked="" type="checkbox"/>	System Administrator	yadav_Vikram
Edit	yadav_Vikram	wada	vp.01012013@gmail.com	11/7/2013 9:00 AM	Manager	<input checked="" type="checkbox"/>	Recruiter	

Then you can find Development Mode and checked check box then click on save button.

After then you can go click on view icon that tin VisualForce ask to create new page because this open in development mode.

Force.com Quick Access Menu

- Development Mode
- Show View State in Development Mode
- Salesforce CRM Content User
- Receive Salesforce CRM Content Email Alerts

Visualforce Error

Page Hom does not exist

 [Create Page Hom](#) ← click on this link create new page

If you need to help or see more component of Salesforce click on component Reference button you can see below.

[Save](#) [Quick Save](#) [Cancel](#) [Where is this used?](#) [Component Reference](#) [Preview](#)

After click on Component Reference button you can see new page on this page you can see below this page.

Visualforce Components

[<apex:actionFunction>](#)

[<apex:actionPoller>](#)

[<apex:actionRegion>](#)

[<apex:actionStatus>](#)

[<apex:actionSupport>](#)

[<apex:areaSeries>](#)

[<apex:attribute>](#)

[<apex:axis>](#)

[<apex:barSeries>](#)

[<apex:canvasApp>](#)

[<apex:chart>](#)

[<apex:chartLabel>](#)

[<apex:chartTips>](#)

[<apex:column>](#)

[<apex:commandButton>](#)

[<apex:commandLink>](#)

[<apex:component>](#)

[<apex:componentBody>](#)

[<apex:composition>](#)

[<apex:dataList>](#)

[<apex:dataTable>](#)

[<apex:define>](#)

`<apex:actionFunction>`

A component that provides support for invoking controller action methods directly from JavaScript code using an AJAX request. An `<apex:actionFunction>` component must be a child of an `<apex:form>` component.

Unlike `<apex:actionSupport>`, which only provides support for invoking controller action methods from other Visualforce components, `<apex:actionFunction>` defines a new JavaScript function which can then be called from within a block of JavaScript code.

Note: Beginning with API version 23 you can't place `<apex:actionFunction>` inside an iteration component — `<apex:pageBlockTable>`, `<apex:repeat>`, and so on. Put the `<apex:actionFunction>` after the iteration component, and inside the iteration put a normal JavaScript function that calls it.

Dynamic Visualforce Notation

Component: Apex.ActionFunction

Component Details

Usage

Attribute Name	Attribute Type	Description	Required?	Default Value	API Version	Accept Formulas?	Access
action	ApexPages.Action	The action method invoked when the actionFunction is called by a DOM event elsewhere in the page markup. Use merge-field syntax to reference the method. For example, action="{!save}" references the save method in the controller. If an action is not specified, the page simply refreshes.			12.0	Yes	global
focus	String	The ID of the component that is in focus after the AJAX request completes.			12.0	Yes	global
id	String	An identifier that allows the actionFunction component to be referenced by other components in the page.			12.0	Yes	global
immediate	Boolean	A Boolean value that specifies whether the action associated with this component should happen immediately, without processing any validation rules associated with the fields on the page. If set to true, the action happens immediately and validation rules are skipped. If not specified, this value defaults to false.			12.0	Yes	global
name	String	The name of the JavaScript function that, when invoked elsewhere in the page markup, causes the method specified by the action attribute to execute. When the action method completes, the components specified by the reRender attribute are refreshed.	Yes		12.0	Yes	global
onbeforedomupdate	String	The JavaScript invoked when the onbeforedomupdate event occurs—that is, when the AJAX request has been processed, but before the browser's DOM is updated.			12.0	Yes	global
oncomplete	String	The JavaScript invoked when the result of an AJAX update request completes on the client.			12.0	Yes	global
rendered	Boolean	A Boolean value that specifies whether the component is rendered on the page. If not specified, this value defaults to true.			12.0	Yes	global

Standard Controllers

A VisualForce controller is a set of instructions that specify what happens when a user interacts with the components specified in associated VisualForce markup, such as when a user clicks a button or link. Controllers also provide access to the data that should be displayed in a page, and can modify component behavior.

if you use the standard Accounts controller, clicking a **Save** button in a VisualForce page results in the same behavior as clicking **Save** on a standard Account edit page.

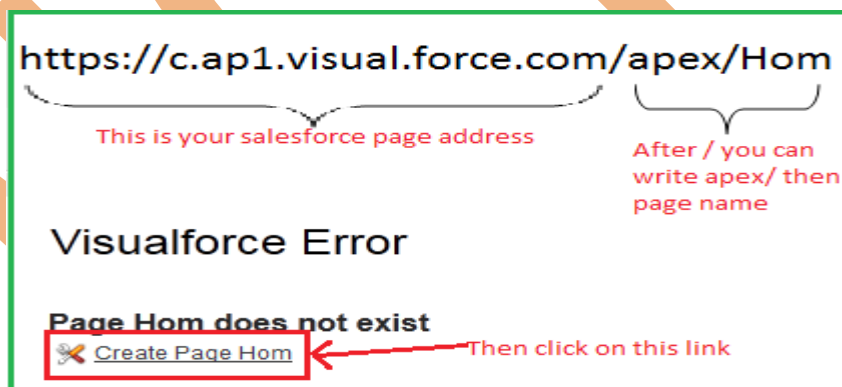
Example :- Create Accounts Standard Controllers.

Click on **Setup > develop > Pages**, Then Click on New button.

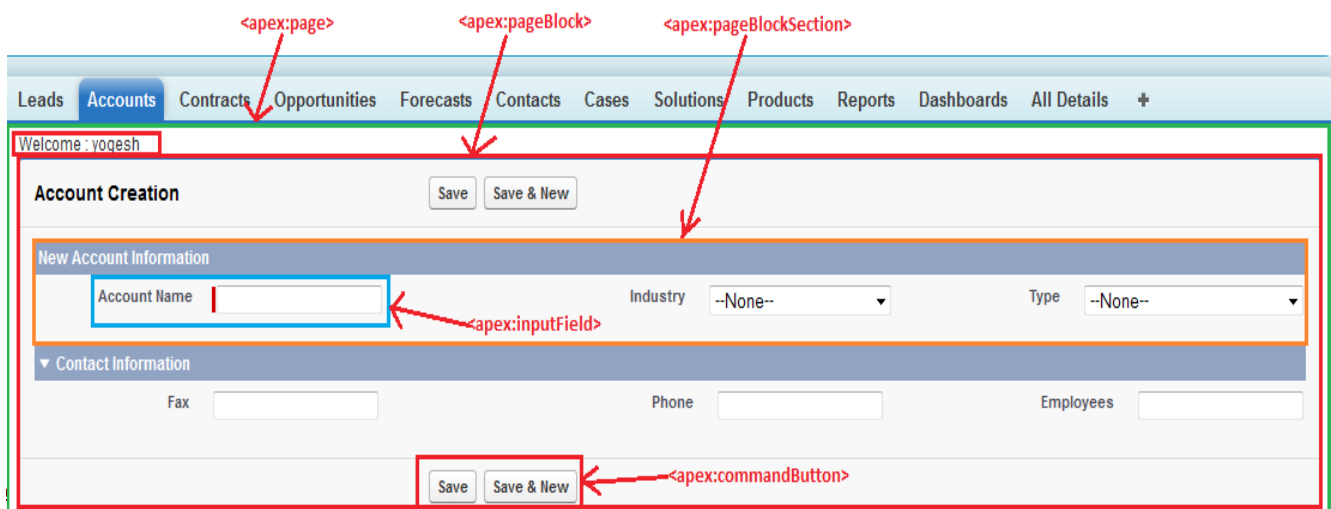
Note :- best way of create pages on SalesForce.

Firstly enable Development Mode Go **Setup > Manage Users > Users** then Click on **Edit** link and checked **Development Mode** checkbox.

After that go to address bar and write `https://c.ap1.visual.force.com/apex/Hom`



Click on Create Page link you can create new page. After click on link you can see new page below on this page designing and coding both part is available in same page you can see below.



```

1 <apex:page standardController="Account">
2   <apex:form >
3     Welcome : {!$User.FirstName}
4     <apex:pageBlock title="Account Creation">
5       <apex:pageBlockSection title="New Account Information" collapsible="false" columns="3">
6         <apex:inputField value="{!account.name}"/>
7         <apex:inputField value="{!account.Industry}"/>
8         <apex:inputField value="{!account.Type}"/>
9       </apex:pageBlockSection>
10      <apex:pageblockButtons >
11        <apex:commandButton value="Save" action="{!save}"/>
12        <apex:commandButton value="Save & New"/>
13      </apex:pageblockButtons>
14    </apex:pageBlock>
15  </apex:form>
16 </apex:page>

```

Position: Ln 10, Ch 5 | Total: Ln 17, Ch 616

Standard Control List :

- Account
- Asset
- Campaign
- Case
- Contact
- Contract
- Idea
- Lead
- Opportunity
- Order
- Product2
- Solution
- User
- Custom objects

Action List In VisualForce :

Action	Description
save	Inserts new records or updates existing records that have been changed. After this operation is finished, the <code>save</code> action returns the user to the original page, if known, or the home page.
quicksave	Inserts new records or updates existing records that have been changed. Unlike

	the <code>save</code> action, <code>quicksave</code> does not redirect the user to another page.
cancel	Aborts an edit operation. After this operation is finished, the <code>cancel</code> action returns the user to the page where the user originally invoked the edit.
first	Displays the first page of records in the set.
last	Displays the last page of records in the set.
next	Displays the next page of records in the set.
previous	Displays the previous page of records in the set.

How to use Action in code :-

```

<apex:page standardController="Account" sidebar="false">
  <apex:form >
    <apex:pageBlock title="Account Creation">
      <apex:pageBlockSection title="New Account Information" collapsible="false"
columns="3">
        <apex:inputField value="{!account.name}"/>
        <apex:pageblockButtons >
          <apex:commandButton value="Save" action="{!save}"/>
        </apex:pageBlock>
      </apex:form>
    </apex:page>

```

Create Records :

```
1 <apex:page standardController="Account">
2   <apex:form >
3     Welcome : {!$User.FirstName}
4     <apex:pageBlock title="Account Creation">
5       <apex:pageBlockSection title="New Account Information" collapsible="false" columns="3">
6         <apex:inputField value="{!account.name}"/>
7         <apex:inputField value="{!account.Industry}"/>
8         <apex:inputField value="{!account.Type}"/>
9         </apex:pageBlockSection>
10        <apex:pageblockButtons >
11          <apex:commandButton value="Save" action="{!save}"/>
12          <apex:commandButton value="Cancel" action="{!Cancel}" />
13        </apex:pageblockButtons>
14      </apex:pageBlock>
15    </apex:form>
16 </apex:page>
```

Position: Ln 14, Ch 26 Total: Ln 16, Ch 633

Home Chatter Campaigns Leads **Accounts** Contracts Opportunities Forecasts Contacts Cases Solutions Products Reports Dashboards All Details +

Welcome : yogesh

Account Creation

New Account Information

Account Name Industry Type

Recent Items

- Cell phone's
- yogesh sharma
- bisp
- my pos

Display Records/Out Field :

```
Newpage1
<apex:page standardController="Opportunity" sidebar="false" recordSetVar="opportunitites">
<apex:form >
<apex:pageblock >
<apex:pageBlockTable value="{!opportunitites}" var="opp">
<apex:column value="{!opp.name}"/>
<apex:column value="{!opp.amount}"/>
<apex:column value="{!opp.CloseDate}"/>
</apex:pageBlockTable>
</apex:pageblock>
</apex:form>
</apex:page>
```

Position: Ln 11, Ch 13 Total: Ln 11, Ch 355

Home Chatter Campaigns Leads Accounts Contracts **Opportunities** Forecasts Contacts Cases Solutions Products Reports Dashboards All Details +

Opportunity Name	Amount	Close Date
Cell phone's	Rs.45,000.00	11/30/2013

Editing Records / Mass Update with List Controllers :

```
1 <apex:page standardController="Opportunity" recordSetVar="opportunities" tabStyle="Opportunity" sidebar="false">
2   <apex:form >
3     <apex:pageBlock >
4       <apex:pageMessages />
5       <apex:pageBlockButtons >
6         <apex:commandButton value="Save" action="{!save}"/>
7       </apex:pageBlockButtons>
8       <apex:pageBlockTable value="{!opportunities}" var="opp">
9         <apex:column value="{!opp.name}"/>
10        <apex:column headerValue="Stage">
11          <apex:inputField value="{!opp.stageName}"/>
12        </apex:column>
13        <apex:column headerValue="Close Date">
14          <apex:inputField value="{!opp.closeDate}"/>
15        </apex:column>
16      </apex:pageBlockTable>
17    </apex:pageBlock>
18  </apex:form>
19 </apex:page>
```

Position: Ln 15, Ch 21 Total: Ln 19, Ch 845

There is you can see out put .

The screenshot shows the Salesforce user interface for the 'Opportunities' tab. At the top, there is a navigation bar with tabs for Home, Chatter, Campaigns, Leads, Accounts, Contracts, Opportunities (selected), Forecasts, Contacts, Cases, Solutions, Products, Reports, Dashboards, and All Details. Below the navigation bar, there is a 'Save' button. The main content area displays a table with the following structure:

Opportunity Name	Stage	Close Date
Cell phone's	Prospecting	11/30/2013 [11/14/2013]

At the bottom of the table, there is another 'Save' button.