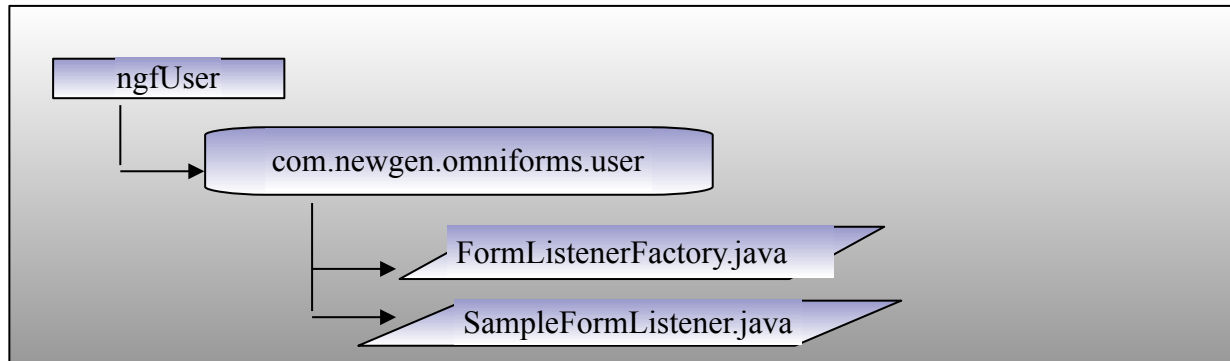


1) Custom Coding in Java

First Configure the Project Sample Code Project in IDE.
Structure of Code Project shown in below diagram.



FormListenerFactory.java:

This file is the initiation point of execution flow. It will redirect control to the class file of particular class. It is highly recommended to use this class as it is without any change. This class implements IFormListenerFactory interface.

This class fetch object on the basis of any condition. Condition could be applied on the basis of General Data Attributes like ActivityName, ProcessName etc and redirect the execution flow to that file.

SampleFormListener.java:

These classes contain custom code and added into workspace by the implementation developers. These classes must either implement FormListener or need to implement all the methods of FormListener Interface.

How framework works:

1. Form passes execution to FormListenerFactory.
2. FormListenerFactory return object of class having implementation logic.

Points to remember:

- Add omniforms.jar library to application for Custom Code Implementation.(Always refer **omniforms.jar** from **webdesktop.war/WEB-INF/lib** folder)
- Add JSF 2.0 library to application for Custom Code Implementation.
(libraries are **jsf-api.jar** and **jsf-impl.jar**. Both are present in webdesktop.war/WEB-INF/lib folder)
- Don't modify any of the supplied packages.
- Implement getListener method in FormListenerFactory.java.
- Implement FormListener interface in classes implementing business logic.
- Import FacesContext package to fetch FormReference and FormConfig Object.
- Jar name could be anything. But structure should be as defined above.
- Need to deploy jar file at **webdesktop.war/WEB-INF/lib** folder.

Note : Use JDK 1.6 to compile the source code.

2)Process Specific Custom Coding in Java

There are two ways to achieve this:

- a) In FormListenerFactory.java file we can check process name from FormConfig Object.

```
FormConfig objFormConfig = FormContext.getCurrentInstance().getFormConfig( );  
String processName = objFormConfig.getConfigElement("ProcessName");
```

Based on processName we can return the instance of particular Java Class file based on condition. For ex : If one user is working on process P1 and other process is working on process P2 then they can return the instance of different class file say for P1 process person is doing code in P1.java and for P2 process Java File is P2.java.

Then condition can be checked in getListener method as shown below:

```
@Override  
public FormListener getListener() {  
    FormConfig objFormConfig = FormContext.getCurrentInstance().getFormConfig( );  
    String processName = objFormConfig.getConfigElement("ProcessName");  
  
    if( processName.equals("P1") )  
        return new P1();  
    else if( processName.equals("P2") )  
        return new P2();  
    else  
        return null;  
  
}
```

- b) Rename FormListenerFactory.java to processName.java and define getListener method in that file. Rest of coding guideline remain same.

To define Multiple process Specific file, create another java file with another processName.java. Similarly we can have activity specific Java Files. For this make the condition on basis of activityName as shown below:

```
FormConfig objFormConfig = FormContext.getCurrentInstance().getFormConfig( );  
String activityName = objFormConfig.getConfigElement("ActivityName");
```

3) Execute FormBuilder API in Java

To Call FormBuilder API's we need to create FormReference Object as shown below:

```
FormReference formReference = FormContext.getCurrentInstance().getFormReference();
```

Note :

1. **We need to define FormReference object in each product provided method in custom coding File.**
2. **Do not Declare it globally anywhere in any of the class files.**
3. **Do not create instance or declare this object in FormListenerFactory Class.**

After declaring FormReference Object We are able to call any product API through this object. For ex : If we want to fetch Values from some TextField say Text1, we can write code as shown below.

```
formReference.setNGValue("Text1", "Mohit");
```

Similarly to get Value from from TextField we can write :

```
formReference.getNGValue("Text1");
```

API List and Use Cases are provided in later section of this document.

4) Get GeneralData Field in Java

As mentioned earlier in section 1 we can get processName from FormConfig object. Similarly we can get any GeneralData Field like activityName, ProcessInstanceId, WorkItemId, CabinetName, JTS IP & Port etc from FormConfig object as shown below.

```
FormConfig objFormConfig = FormContext.getCurrentInstance().getFormConfig( );  
String processName = objFormConfig.getConfigElement("ProcessName");  
String activityName = objFormConfig.getConfigElement("ActivityName");  
String cabinetName = objFormConfig.getConfigElement("EngineName");
```

5) Usage of FormLoaded Method

If User wants to Initialize some control fields or wants to fetch data from master table, he can use formloaded method. This method is called when the form gets rendered on the browser but the data from the external table has not set on the control field yet. So value saved in external table will not be available in this method. API Calling guideline is same as shown earlier. (comments shown in green color):

```
@Override  
public void formLoaded(FormEvent pEvent) {  
// Need to declare this object in each and every method  
FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
}
```

6) Usage of FormPopulated Method

If User wants to Initialize some control fields or wants to fetch data from master table on basis of values from external table, user can write code in this method. This method is called when the form gets rendered on the browser and data from the external table has been set on the control field. So we can get the value of any control which has set previously in this method. API calling guideline is same as shown earlier.

@Override

```
public void formPopulated(FormEvent pEvent) {  
  
    // Need to declare this object in each and every method  
    FormReference formReference = FormContext.getCurrentInstance().getFormReference();  
    formReference.setNGListIndex("qvar_PersonalandFamilyInfo", 0);  
}
```

7) Usage of EventDispatched Method

When any event gets triggered from form like value change event of any field or mouse clicked event of button, then this method is called when server side coding is enabled from javascript. (To enable server side coding in js for particular control refer section 12). code snippets to write code in event dispatched method is shown below.

```
public void eventDispatched(ComponentEvent pEvent) throws ValidatorException  
{  
    FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
  
    switch (pEvent.getType()) { // Check the type of event first  
        case FRAME_COLLAPSED : {  
            formObject.setNGValue("NB_BILLING_FREQUENCY", "rohit");  
            break;  
        }  
        case MOUSE_CLICKED: {  
            // pEvent.getSource().getName() returns the name of the control which triggered  
            // the event for ex: on click of button say button1 this will return name of button Button1  
            if(pEvent.getSource().getName().equalsIgnoreCase("Button1"))  
            {  
                formObject.setNGForeColor("DatePicker1", Color.RED);  
            }  
            break;  
        }  
        case VALUE_CHANGED: {  
            if(pEvent.getSource().getName().equalsIgnoreCase("DatePicker4"))  
            {  
                formObject.setNGFocus("CP_CREATIONDATE");  
            }  
            break;  
        }  
        default : { break;  
    }  
}
```

8) Usage of saveFormStarted Method

User can write code in this method If User wants to perform certain validation prior to saving of form. API calling guideline is same as shown earlier.

```
@Override  
public void saveFormStarted(FormEvent pEvent) throws ValidatorException  
{  
    // Need to declare this object in each and every method  
    FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
}
```

9) Usage of submitFormStarted Method

User can write code in this method If User wants to perform certain validation prior to Completion of workitem (Introduce or Done Event). API calling guideline is same as shown earlier.

```
@Override  
public void submitFormStarted(FormEvent pEvent) throws ValidatorException  
{  
    // Need to declare this object in each and every method  
    FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
}
```

10)Usage of saveFormCompleted Method

User can write code in this method If User wants to give success message (How to show message will be covered in section 15) for saving of form in this method. API calling guideline is same as shown earlier.

```
@Override  
public void saveFormCompleted(FormEvent pEvent) {  
    // Need to declare this object in each and every method  
    FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
}
```

11)Usage of submitFormCompleted Method

User can write code in this method If User wants to give success message (How to show message will be covered in section 15) after performing Introduce/Done of WorkItem in this method. API calling guideline is same as shown earlier.

```
@Override  
public void submitFormCompleted(FormEvent pEvent) throws ValidatorException {  
    // Need to declare this object in each and every method  
    FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
}
```

12)Custom Coding in Js

To enable server side coding first we need to write code in eventDispatched method in client.js. This method is called for any event trigger from form like mouse click event, value changed event, focus event , blur event. FormBuilder JavaScript API's can be called using below code snippet.

```
com.newgen.omniforms.formviewer.getNGValue("Text1");
```

Here *com.newgen.omniforms.formviewer* is the package strcuture and **getNGValue** is the name of the API. To enable server side coding for any event **return true** corresponding to that event or return false otherwise. Method Signature and sample code snippet for writting code is shown below:

Note : It is strictly recommended to return false by default from this method. Otherwise every event triggered will be propagated to server and performance will be hindered.

In this method **pId** will be the name of the control on which event is triggered and **pEvent** will be the event. Third parameter will signify whether the frameState is collapsed or Expanded(0 for expanded 1 for collapsed). User can write code on basis of event type and control name as shown below:

```
function eventDispatched(pId,pEvent,collapseornot){  
switch(pEvent.type)  
{  
case 'click':  
{  
    if( pId == "Button1")  
        com.newgen.omniforms.formviewer.setNGForeColor("DatePicker1","yellow");  
        return true; // Server side eventDispatched method will be called  
    }  
case 'change': {  
    if( pId == "q_apnlines_amount" ){  
        com.newgen.omniforms.formviewer.setNGFrameState("Frame6",0);  
        return false. // Server side eventDispatched method will not be called  
    }  
    if(pId == "RM_NAME"){  
        return true; // Server side eventDispatched method will be called  
    }  
    if( pId == "BRANCH_NAME"){  
        com.newgen.omniforms.formviewer.setNGFrameState("Frame4",1);  
        return true;// Server side eventDispatched method will not be called  
    }  
    break;  
    }  
case 'keydown':{  
    return false;  
    break;  
    }  
case 'focus':{  
    break;  
    }  
}  
return false; // by default it should return false  
}
```

13)Process Specific Custom Coding in Js

User can call custom method based on processName. Process name can be get in Javascript using below code snippet.

```
window.parent.strprocessname // this will give name of the current process
```

for example to called processbased custom code we can write code as shown below

```
function eventDispatched(pId,pEvent,collapseornot){
    var processName = window.parent.strprocessname;
    if( processName == "LoanProcess"){
        return eventDispatchedLoanProcess(pId,pEvent,collapseornot)
    }
    else if( processName == "AccountPayable"){
        return eventDispatchedAccountPay(pId,pEvent,collapseornot)
    }
}
```

In this case user needs to define own custom JS method.

14)Get GeneralData Fields in Js

```
window.parent.strprocessname // name of current process
window.parent.stractivityName // name of activity
window.parent.strRouteId // current routeid
window.parent.strqueueType // queue type
window.parent.stractivityType // activity type
window.parent.stractivityId // activityid
```

15)Show UI Alert from Java

To show customized alert from Java use below code snippet.

Here Message **Date cannot be Current or future or less than 18 Years** will be shown as alert after pressing ok focus will gets shifted to **DatePicker1** control.

```
throw new ValidatorException(new FacesMessage("Date cannot be Current or future or less than 18 Years","DatePicker1"));
```

16)Show UI Alert form Js

Similarly to show Alert from Js we can write below code snippet :

```
com.newgen.omniforms.util.showError("DatePicker1","Date cannot be Current or future or less than 18 Years")
```

17)Configure Product Specified Alert Message from Js

To change the message of product shown alert, user need to implement code as shown below:

```
function getMessage(controlName,message){
    if( controlName == "ListView1" ){
        return "Table can Not be empty";
    }
}
```

In this case, customized message "Table can Not be empty" will be shown to user instead of product message.

18)Show Confirm Box from Java

From Java User Needs to throw ValidatorException as shown below.

Note : Do not put below statement in try catch.

```
HashMap hm ; // not nullable HashMap
throw new ValidatorException( new CustomExceptionHandler( "Error Message", "Text1",
"EventType", hm ));
```

In above scenario the parameter passed from Java can be collected in Javascript as shown below in client.js. If below method is not defined in client.js, please define it.

```
function customEventHandler(excpHandlerJson)
{
    var eventName = expHandlerJson.EventHandler;
    // 3rd parameter from CustomExceptionHandler constructor defined in java
    var params = expHandlerJson.Parameters;
    // 4th parameters which is hashmap in form of JSON from CustomExceptionHandler
    constructor defined in java
    var firstParam = expHandlerJson.Summary;
    //1st parameter from CustomExceptionHandler constructor defined in java
    var secondParam = expHandlerJson.Detail;
    //2nd parameter from CustomExceptionHandler constructor defined in java

    // custom coding goes here

    return expHandlerJson;
}
```


19)Custom Validation in Js

User can write code in validateForm method in client.js as shown below:

Note : pEvent can be S (For Save Operation),D(For Done Event),I (For Introduce).

return true when validation successful else return false.

```
function validateForm(pEvent)
{
    // write custom code here
    return false;
}
```

20)Validation on Save/Submit in Java

User can throw ValidatorException from SaveFormStarted/SubmitFormStarted as discussed earlier.

21)Validation on Save/Submit in Js

Please refer section 19.

22)Set Style from Java Code

As mentioned earlier in this document that to call Java Api we need to create **FormConfig** object then call the method described below.

22.1)Set Font Color of Control

Here is the code snippet to with following API call, user can use any of the following API calls to set Font color of Control ,here in following examples we have taken Text1 .

```
formReference.setNGForeColor("Text1", Color.yellow);  
formReference.setNGForeColor("Text1", value);  
formReference.setColor("Text1" , Color.yellow);
```

Here **Text1** is name of control and **Color.yellow** is color user want to set.
Value is integer value of color which user want to set.
In above example yellow has been chosen as color .

22.2) Set Font Size of Control

Here is the code snippet to with following API call, user can use any of the following API calls to set Font Size of control . Here we are setting size of Text1 control to 13.

```
formReference.setNGControlFont("Text1", "Arial Black", 13);  
formReference.setNGControlFont("Text1", "Arial Black", 13, true, true);
```

Here in second API call we set font size 13 and font family Arial Black and also giving other two flag **Bold** and **Italic** as true value.

22.3) Set Font Family of Control

Here is the code snippet to with following API call, user can use any of the following API calls to set Font Size of control . Here we are setting size of Text1 control to 13.

```
formReference.setNGControlFont("Text1", "Arial Black", 13);  
formReference.setNGControlFont("Text1", "Arial Black", 13, true, true);
```

Here in second API call we set font size 13 and font family Arial Black and also giving other two flag **Bold** and **Italic** as true value.

```
formReference.setFont("Text1", fontObject);
```

This API set a particular font on the control Text1.
Note here fontObject is java representation of the Font.

22.4) Set Font Weight of Control

Here is the code snippet with following API call, to set font bold.

```
formReference.setNGControlFont("Text1", "Arial Black", 13, true, true);
```

Here in API call we set font size 13 and font family Arial Black and also giving setting two flag **Bold** and **Italic** as true value.

22.5) Set Background Color of Control

This API call we use to set Background color of any control.

```
formReference.setNGBackColor("Text1", Color.yellow);
```

Here in above API call we have change background color of **Text1** control to **yellow**.

```
formReference.setNGBackColor("Text1", integerValueOfColor);
```

Here in above API we give control Name and Integer value of a color to this api ,it will set the back ground color according.

22.6) Set Top of Control

This API call is used to set the Top of control . Here is the example

```
formReference.setTop("Text1", 20);
```

In the above API call it will set the top of control **Text1** to 20.

22.7) Set Left of Control

This API call can be used to set Left of Control. Here is the example explaining the use of this API .

```
formReference.setLeft("Text1", 125);
```

Here is first parameter in this API is control name in above example it is Text1 and an integer value which we want to set as left of the control. Here 125 is the value of left.

22.8) Set Width of Control

This API call explain the usage to set the width of a control.

```
formReference.setWidth("Text1", 130);
```

Above API call will set width of control Text1 to 130 pixels .

22.9) Set Height of Control

This API call is to change the Height of any control .

```
formReference.setHeight("Text1", 130);
```

Here in above example the API changes or set the height of control Text1 to 130 pixel .

Note : In both the width and height user has to give integer to second parameter.

22.10) Enable/Disable Control

This API call example to enable or disable a component.

```
formReference.setEnabled("Text1", true);
```

Above example of API is to enable a component/control Text1.

```
formReference.setEnabled("Text1", false);
```

Above example of API is to disable a component/control Text1.

22.11) Lock (ReadOnly)/Unlock Control

Here is an example of API call explaining the use to Lock and Unlock a control.

```
formReference.setLocked("Text1", true);
```

Here above call lock a control name **Text1**.

```
formReference.setLocked("Text1", false);
```

Here above call is used to unlock a control text1 which is previously locked.

22.12) Visible/Invisible of Control

This API call is used to set a control visible or invisible .

```
formReference.setVisible("Text1", true);
```

above call set text1 control visible if it previously not visible.

```
formReference.setVisible("Text1", false);
```

above API call will make a control named text1 invisible.

22.13) Set Alignment of Control

Following API call explaining the usage for alignment of text of Control.

```
formReference.setTextAlignment("Text1", "right");
```

Here in above example using this API call we can set text alignment of control **Text1** to **right**. Possible values are **left, right, center**.

22.14) Set CharCase of TextField Control

This API call can be used to set the character case of text of control .

formReference.setNGCharCase("Text1", 'U');

Here above example call is to set character case to UPPER case for control Text1.

formReference.setNGCharCase("Text1", 'L');

Here above API call explaining the usage for the Lower case.

formReference.setNGCharCase("Text1", 'N');

Here above api call set the character case to default.

23)Set Style from Js Code

23.1) Set Font Color of Control

This JS API is used to set the Color of fonts of Control.

com.newgen.omniforms.formviewer.setNGForeColor("Text1", "red");

Here in above call to API we pass two paramere first Id of element or control second second parameter is color we want to set.

23.2) Set Font Size of Control

Following API call is being used to set the size of the font .

com.newgen.omniforms.formviewer.

setNGControlFont("Text1", "Arial Black", 13, true, true);

Here in above call to API Text1 is the id of the control and Arial Black is font and 13 is the size of the font and with this there are two flag for **Bold** and **Italic**.

23.3) Set Font Family of Control

This API is being used to set the font-family of the font .

com.newgen.omniforms.formviewer.

setNGControlFont("Text1", "Arial Black", 13, true, true);

Here in above call to API Text1 is the id of the control and Arial Black is font and 13 is the size of the font and with this there are two flag for **Bold** and **Italic**.

23.4) Set Font Weight of Control

This API is used to set the Weight of the font .

```
com.newgen.omniforms.formviewer.  
    setNGControlFont("Text1", "Arial Black", 13, true, true);
```

Here in above call to API Text1 is the id of the control and Arial Black is font and 13 is the size of the font and with this there are two flag for **Bold** and **Italic**.

23.5) Set BackGround Color of Control

This JS API is being used to set the Background-Color of Control.

```
com.newgen.omniforms.formviewer.setNGBackColor("Text1", "red");
```

Here in above call to API we pass two paramere first Id of element or control second second parameter is color we want to set.

23.6) Set Top of Control

Following API call use to set the Top of control . Here is the example

```
com.newgen.omniforms.formviewer.setTop("TextBox1", 20);
```

In the above API call it will set the top of control with Id TextBox1 to 20.

23.7) Set Left of Control

Following API call can be used to set Left of Control. Here is the example explaining the use of this API .

```
com.newgen.omniforms.formviewer.setLeft("Text1", 125);
```

Here is first parameter in this API is control name in above example it is Text1 and an integer value which we want to set as left of the control. Here 125 is the value of left.

23.8) Set Width of Control

This API call explain the usage to set the width of a control.

```
com.newgen.omniforms.formviewer.setWidth("Text1", 130);
```

Above API call will set width of control with Id Text1 to 130 pixels .

23.9) Set Height of Control

This API call is used to change the Height of any control.

```
com.newgen.omniforms.formviewer.setHeight("Text1", 130);
```

Here in above example the API changes or set the height of control with Id Text1 to 130 pixel .

Note : In both the width and height user has to give integer value to second parameter.

23.10) Enable/Disable Control

This API call is used to enable or disable a component.

```
com.newgen.omniforms.formviewer.setEnabled("Text1", true);
```

Above example of API is to enable a component/control with Id Text1.

```
com.newgen.omniforms.formviewer.setEnabled("Text1", false);
```

Above example of API is to disable a component/control with Id Text1.

23.11) Lock (ReadOnly)/Unlock Control

Following API call explain the use to Lock and Unlock a control

```
com.newgen.omniforms.formviewer.setLocked("text1", true);
```

Here above call lock a control with id **text1**.

```
com.newgen.omniforms.formviewer.setLocked("text1", false);
```

Here above call is used to unlock a control with id text1 which is previously locked.

23.12) Visible/Invisible of Control

This the API call which is being used to set a control visible or invisible .

```
com.newgen.omniforms.formviewer.setVisible("Text1", true);
```

above call set control with id text1 visible if it previously not visible.

```
com.newgen.omniforms.formviewer.setVisible("Text1", false);
```

above API call will make a control with id text1 invisible.

23.13) Set Alignment of Control

This API call will be used for alignment of text of Control.

```
com.newgen.omniforms.formviewer.
```

```
SetNGTextHorizontalAlignment("textBox1", "right");
```

Here in above example using this API call we can set text alignment of control with id **textBox1** to **right**.

23.14) Set CharCase of TextField Control

This is the API call can be used to set the character case of text of control with Id text1 .

```
com.newgen.omniforms.formviewer.setNGCharCase("text1", 'U');
```

Here above example call is to set character case to UPPER case for control text1.

```
com.newgen.omniforms.formviewer.setNGCharCase("text1", 'L');
```

Here above API call explaining the usage for the Lower case.

```
com.newgen.omniforms.formviewer.setNGCharCase("text1", 'N');
```

Here above api call set the character case to default.

24)Get Style from Java Code

As mentioned earlier in this document that to call Java Api we need to create **FormConfig** object then call the method described below.

24.1)Get Font Color of Control

Here is the code snippet to with following API call, user can use any of the following API calls to get Font color of Control ,here in following examples we have taken **text1** .

```
formReference.getNGForeColor("text1");
```

Here **text1** is name of control for which we want get Color property.

24.2) Get BackGround Color of Control

This API call demonstrating how to get Background color of a control

```
formReference.getBackcolor("TextBox");
```

To this API method we pass the name of the control to get the value of the background color.This API call will return a color object instance of java class Color.

24.3) Get Top of Control

This API call is used to get the Top of a control . Here is an example

```
formReference.getTop("TextBox1");
```

In the above API call it will return the top of control TextBox1.

24.4) Get Left of Control

This API call can be used to get Left of Control. Here is the example explaining the use of this API .

```
formReference.getLeft("Text");
```

Here in this API we pass name of the control and it will return the left of corresponding control.

24.5) Get Width of Control

This API call explains the usage to get the width of a control.

```
formReference.getWidth("textBox");
```

Above API call will return width of control **textBox** .

24.6) Get Height of Control

This is the API call used to get height of any control .

formReference.getHeight("Text1");

Here in above example the API changes or get the height of control Text1.

24.7) Check Control Visibility

This API call will be used to get a control's visibility .

formReference.isVisible("Text1");

above call return the visibility status of Text1 control .

24.8) Check Control is Locked

This is the API call used to get Locked status of a control.

formReference.isLocked("Text2");

Here above call to API will return the status of control's status of Lock.

24.9) Check Control Enabled or not.

This API call can be used to check whether control is enabled or not .

formReference.isEnabled(null);

Above call to API will return the true/false depending upon whether control is enabled/disabled.

25)Set Style from Js Code

Here is the guide for using JS API of Form Builder .

25.1) Get Font Color of Control

This API call is used to get the color of the control with id.

```
com.newgen.omniforms.formviewer.getNGForeColor("Text1");
```

This API will return the color code in the form Hexadecimal Code.

25.2) Get BackGround Color of Control

Following API call is being used to get the Background of the control with id Text1.

```
com.newgen.omniforms.formviewer.getNGBackColor("Text1");
```

This API will return the Background Color code in the form Hexadecimal Code.

25.3) Get Top of Control

This API call will be used to get the Top of the control with id.

```
com.newgen.omniforms.formviewer.getTop("Text1");
```

This API will return the top of the control with Text1.

25.4) Get Left of Control

Following API call will be used to get the Left of the control with id Text1.

```
com.newgen.omniforms.formviewer.getLeft("Text1");
```

This API will return the left of the control with Text1.

25.5) Get Width of Control

This API call is being used to get the width of the control with id Text1.

```
com.newgen.omniforms.formviewer.getWidth("Text1");
```

This API will return the width of the control with Text1

25.6) Get Height of Control

This API call is used to get the height of the control with id Text1.

com.newgen.omniforms.formviewer.getHeight("Text1");

This API will return the height of the control with Text1.

25.7) Check Control Visibility

This API call is used to check the visibility of the control with id Text1.

com.newgen.omniforms.formviewer.isVisible("Text1");

This API will return the height of the control with Text1.

25.8) Check Control is Locked

This API call is used to check the Locked of the control with id Text1.

com.newgen.omniforms.formviewer.isLocked("Text1");

This API will return the status of Locked of the control with id Text1.

25.9) Check Control Enabled or not

This API call is used to check the Visibility of the control with id Text1.

com.newgen.omniforms.formviewer.isLocked("Text1");

This API will return the status of Visibility of the control with id Text1.

26) Set Value in Control from Java Code

Note: Here to use all the Java related API we need to create object of **FormReference** as shown here below.

```
FormReference formReference =  
FormContext.getCurrentInstance().getFormReference();
```

Here is the code snippet showing how to use this API to set the value of a control.

```
formReference.setNGValue("name", "Mohit");
```

This will set the value Mohit in control name .

```
formReference.setNGValue("text1", "yourValue", true);
```

This API call will set the value yourValue in control text1 and set the render flag of the control to true.

27) Set Value in Control from Js Code

This API call is used to set the value of the control with id Text1.

```
com.newgen.omniforms.formviewer.setValue("Text1","valueObject");
```

This API will set the value equal to valueObject in the control with id Text1.

28) Get Value of Control from Java Code

Here is the code snippet showing how to use this API to get the value of a control.

```
formReference.getNGValue("text1");
```

This will return the value of control name text1 .

```
formReference.getNGValue("listView1", 2, 2);
```

This API call will return the value of control listView1 at row no. 3 and column 3.

Note: in java index is counted with start from 0.

29) Get Value of Control from Js Code

This API call is used to get the value of the control with id Text1.

```
com.newgen.omniforms.formviewer.getNGValue("Text1");
```

This API will return the value of control with id Text1.

This API call is used to get the value of the control with id Text1 and index.

```
com.newgen.omniforms.formviewer.getNGValueAt("Text1",5);
```

This API will return the value of control with id Text1 and at index 5 of control.
This API is used to get the value if control is of list type .

This API call is used to get the value of the control with id Text1 and index.

```
com.newgen.omniforms.formviewer.getValueAt("Text1",5);
```

This API will return the value of control with id Text1 and at index 5 of control.
This API is used to get the value if control is of list type .

30) Clear Value of Control from Java Code

Here is the example explaining the usage of this API.

```
formReference.clear("text1");
```

This will clear the value of control name **text1** .

31) Clear Value of Control from Js Code

This API call is used to clear the value of the control with id Text1.

```
com.newgen.omniforms.formviewer.NGClear("Text1");
```

This API will clear the value of the control with id Text1.

32) Set ToolTip on Control from Java Code

Here is the example explaining the usage of this API.

```
formReference.setToolTip("text1", "Name of Candidate");
```

This will set value of tool tip to **Name of Candidate** for the control with name **text1** .

33) Set ToolTip on Control from Js Code

This API call is used to set the value of the tool tip for control with id Text1.

```
com.newgen.omniforms.formviewer.setToolTip("Text1","Name of Employee");
```

This API will set the value of tool tip to **Name of Employee** for control with id Text1.

34) Remove ToolTip from All Control in Java

Here is the API used to remove the tool tip from the Form.

```
formReference.removeNGTooltip();
```

This will remove tool tip from all the controls.

35) Set Frame Collapsible/Expand from Java

Here is the code snippet showing how to use this API to set the frame state.

```
formReference.setNGFrameState("Frame1", 0);
```

This call will set the state of the frame to expanded state for the frame name **Frame1**.

```
formReference.setNGFrameState("Frame1", 1);
```

This call will set the state of the frame to collapsed state for the frame name **Frame1**.

36) Set Frame Collapsible/Expand from Js

This API call is used to change the state of the frame with id **Frame1**.

```
com.newgen.omniforms.formviewer.setNGFrameState("Frame1",0);
```

This call will set the state of the frame to expanded state for the frame with id **Frame1**.

```
com.newgen.omniforms.formviewer.setNGFrameState("Frame1",1);
```

This call will set the state of the frame to collapsed state for the frame with id **Frame1**.

37) Set FloatFormatting of Particular Float Control in Java

Here is the API call explaining the usage of this API .

```
//formReference.NGLocalFloatFormat(null, iTotalDigitsInDec, iDigitAfterDec);  
formReference.NGLocalFloatFormat("floatText", 10, 3);
```

As it is obvious from the second call to method explaining that there will be total 10 digit in the no. And after decimal there will be 3 digits .

38) Set FloatFormatting of All Float Fields in Java

Here is the example explaining the usage of this API.

```
formReference.NGGlobalFloatFormat(10, 3);
```

This API call will set the formatting for all the float fields . It will set the total digits to 10 and digits after the decimal to 3 for all the float field in form.

39) Set Date Range (Min/Max) From Java

Here is the API call explaining the use of this .

```
formReference.setNGDateRange("DatePicker1",new Date("2015/07/10"), new  
Date("2015/07/15"));
```

This call will set the DatePicker1's min date and maximum date as corresponding mention above used in call.

40) Add Items in Combo Box from Java using ItemLabel (value will be same as label)

This API call is used to set the value of the combobox control with name combobox1.

```
formReference.addItem("combobox1", "ItemOne");
```

This API will set the value and label to **ItemOne** for control combobox with name combobox1.

41) Add Items in Combo Box from Java using ItemLabel and ItemValue

This API call is used to set the value of the combobox control with name combobox1.

```
formReference.addItem("combobox1", "ItemOne","ValueOne");
```

This API will set label to **ItemOne** and the value to **ValueOne** for control combobox with name combobox1.

42) Add Items in Combo Box from Java using ItemLabel,ItemValue and ToolTip

Here is the example explaining the usage of this API.

```
formReference.addItem("combobox1", "ItemOne","ValueOne","This is ValueOne");
```

This API will set label to **ItemOne** and the value to **ValueOne** and tool tip to **This is ValueOne** for control **combo box** with name **combobox1**.

43) Add Items in Combo Box from Js using ItemLabel (value will be same as label)

This API call is used to set the value of the combobox control with id combobox1.

```
com.newgen.omniforms.formviewer.addItem("combobox1","ItemOne");
```

This API will set the value and label to **ItemOne** for control combobox with id combobox1.

44) Add Items in Combo Box from Js using ItemLabel and ItemValue

This API call is used to set the value of the combobox control with id combobox1.

```
com.newgen.omniforms.formviewer.addItem("combobox1","ItemOne","ValueOne");
```

This API will set label to **ItemOne** and value to **ValueOne** for control combobox with id combobox1.

45) Remove Item from Combo Box in Java

Here is the API call explaining the use of this API .

```
formReference.removeItem("Combobox1",2);
```

This will remove the Item at index 2 from the combobox with name **Combobox1** .

46) Set Alignment in Combo Options in Java

Here is the example explaining the usage of this API.

```
formReference.setComboAlignment("combobox1", 20);
```

This API will move the items of the combobox towards the right with 20px.

47) Get Item Count of Control (Combo/ListBox)

Following example explain the usage of this API.

```
formReference.getItemCount("combobox1");
```

This method will return the count of the combobox with name **combobox1**. Similarly this can be used for the ListBox.

48) SetFocus on Control from Java

Here is the explanation of using this API with Example.

```
formReference.setNGFocus("combobox1");
```

This will set focus on the combobox with name **combobox1**.

This can be used for any control in the form.

49) Set Focus on Control from Js

This API call is used to set focus on the control with id combobox1.

```
com.newgen.omniforms.formviewer.setNGFocus("combobox1");
```

This will set focus on the combobox with id **combobox1**.

This can be used for any control in the form.

50) Set Frame Locked in Java

Here is the explanation of using this API with Example.

```
formReference.setFrameLockedWithCollapsed("frame1",true,0);
```

This will lock the frame with expanded state . Here second parameter determines the lock status of the **frame1** and last parameter determines whether **frame1** is expanded or collapsed or not depending upon the value passed.

51) Set Frame Locked in JS

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.setFrameLockedWithCollapsed("frame1",0,true);
```

This will lock the frame with expanded state . Here third parameter determines the lock status of the **frame1** and second parameter determines whether **frame1** is expanded or collapsed or not depending upon the value passed.

52) Set Sheet of Tab Visible/Invisible from Java

Here is the explanation of using this API with Example.

```
formReference.setSheetVisible("frame1",2,true);
```

Here this API will set tab sheet with tab index 2 visible. The last parameter determines whether tab sheet to be made visible or not.

53) Set a particular Sheet Enable/Disable

Here is the explanation of using this API with Example.

```
formReference.setSheetEnable("frame1",2,true);
```

Here this API will set tab sheet with tab index 2 enable. The last parameter determines whether tab sheet to be made enable or disable.

54) Make Tab Multiline in Java

Here is the explanation of using this API with Example.

```
formReference.setNGMultilineTab("tab1",true);
```

Here this API will set the Tab control multiline with name **tab1**.

55) Set Particular Sheet Selected of Tab in Java

Here is the explanation of using this API with Example.

```
formReference.setSelectedSheet("tab1",2)
```

Here this API will set selected sheet with index 2 for Tab control with name **tab1**.

56) Get Selected Sheet Index of Tab in Java

Here is the explanation of using this API with Example.

```
formReference.getSelectedSheet("tab1")
```

Here this API will return index of the selected sheet for the control with name **tab1**.

57) Add Row in ListView from Java

Here is the explanation of using this API with Example.

```
formReference.NGAddListItem("SecDFollowupGrid",  
"<ListItem><SubItem>1</SubItem><SubItem>2</SubItem><SubItem>3</SubItem>  
</ListItem>");
```

Here this API will add a Row in the **SecDFollowupGrid**.

Following API also can be used to add a Row in ListView

```
formReference.ExecuteExternalCommand("NGAddRow", "listView1");
```

Above API call will add a row in **listView1**.

58) Modify Row in ListView from Java

Following example explains the usage of this API.

```
formReference.ExecuteExternalCommand("NGModifyRow", "listView1");
```

Above API call will modify the selected rows in **listView1**.

59) Delete Row in ListView from Java

Here is the explanation of using this API with Example.

```
formReference.removeSelectedRows("ListView1");
```

Above API will remove a selected Row from control having name **ListView1**.

This is another API that can be used to Delete row from a ListView.

```
formReference.ExecuteExternalCommand("NGDeleteRow", "listView1");
```

Above API will remove a selected Row from control having name **listView1**.

60) Insert Row in ListView from Java

This API can be used to insert a Row in ListView

```
formReference.ExecuteExternalCommand("NGAddRow", "listView1");
```

Above API call will insert a row in **listView1**.

61) Get Row Count of ListView in Java

Here is the explanation of using this API with Example.

```
formReference.getLVWRowCount("ListView1");
```

Here this API will return a selected Row from control having name **ListView1**.

62) Get Row Count of ListView in Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.getLVWRowCount("ListView1");
```

This will return the count of the number of rows in the ListView1.

63) Get Value of ListView (XML of all listview fields)

Here is the explanation of using this API with Example.

```
formReference.getNGListView("ListView1");
```

Here this API will return the XML representation of the control name **ListView1**.

64) Set Value in ListView from Table

Here is the explanation of using this API with Example.

```
formReference.getNGValue("queryStringToFetchData", listOfControls);
```

This API call will set data in list of control from database.

65) Get Value present in particular Row & Column of ListView in Java

Here is the explanation of using this API with Example.

```
formReference.getNGValue("listView1", 2, 2);
```

This API call will return the value of control listView1 at row no. 3 and column 3.

Note: in java index is counted with start from 0.

66) Select particular row in ListView

Here is the explanation of using this API with Example.

```
formReference.setSelectedIndex("listView1", 2);
```

This API call will select the row with index 2 in **listView1** control.

67) Select particular rows in ListView

Here is the explanation of using this API with Example.

```
formReference.setSelectedIndices("listView1", {0,1,2,3});
```

This API call will select the row with index **0,1,2,3** in **listView1** control.

Here is the explanation of using this API with Example.

```
formReference.setNGLVWSelectedRows( "listView1", {0,1,2,3} );
```

This API call will select the row with index **0,1,2,3** in **listView1** control.

68) Add Column in ListView

Here is the explanation of using this API with Example.

```
formReference.NGAddColumnHeader("listView1", "newColumn");
```

This API call will add a new column in the list view with header name "**newColumn**" in control with name **listView1** .

69) Modify Column in ListView (Label/Width)

Here is the explanation of using this API with Example.

```
formReference.setNGColumnWidth("listView1",1,125 );
```

This API call will set the width of a column at index 1 to 125 pixel of control **listView1** .

We can use following API to change the label of ListView.

```
formReference.setNGColumnHeaderCaption("listView1", "colName", "newName")
```

This API call will set the **colName** caption to **newName** .

70) Set Visible/Invisible particular column of ListView

Here is the explanation of using this API with Example.

```
formReference.setNGColumnWidth("listView1", 1 , 0);
```

This API call will set the width of a column at index 1 to 0 pixel so the column becomes invisible and similarly to make it visible change its width accordingly for control **listView1** .

71) Set Visible/Invisible Header of ListView

Here is the explanation of using this API with Example.

```
formReference.setLVWHeaderVisible("listView1", true);
```

This API call will make column header visible/invisible according to the second parameter pass as **true/false** for control **listView1** .

72) Set Column Total of particular Column of ListView in Text Field

Here is the explanation of using this API with Example.

```
formReference.setNGColumnTotal("listView1", "col1","text1");
```

This API call will set the total property of a column here in this example **col1** to a control whose name is passed in third parameter, here it is **text1**.

73) Make Any ListView Cell (Row & Column given) Hyperlink and open url on click

Here is the explanation of using this API with Example.

```
formReference.setLVWHyperLink( "listView1",1 ,2,"yourURL");
```

This API call will make a list view cell a hyperlink or a url with link as in third parameter. Here in this example a cell of **listView1** at column idnex 1 and row index 2 will become a url which will be passed to third parameter.

74) Set Sorting Order in ListView

Here is the explanation of using this API with Example.

```
formReference.sortNGListview("listView1", 1, true);
```

This API call will set the sorting order for colum with column index 1 as ascending order because as we have passed third paramter **true** in this API.
In case we want to make it decending order then third parameter will be **false**;

75) Set Particular Rows of ListView disabled

Here is the explanation of using this API with Example.

```
formReference.setLVWRowsDisabled("listView1", {1,2,3,4}, true);
```

This API call will disable the rows with index 1,2,3,4 in listView1.

76) Change Color of particular Cell of ListView in Java

Here is the explanation of using this API with Example.

```
formReference.setNGCellForeColor ( "listView1",1 ,2,Color.Red);
```

Thsi API call will set the background color to red for a cell with rowIndex 1 and cellIndex 2. Similarly we can change the color of other cells by calling this API.

77) Get Cached Data from Table in Java

Here is the explanation of using this API with Example.

```
formReference.getNGDataFromDataCache ( "query","listView1","1,2,3,4");
```

This API call will set fetch data from the query and put the data in the columns of listView1. In this case data will be inserted into the col1,col2,col3,col4.

We can also use following API to fetch data on precondition.

```
formReference.getNGDataFromDataCache( "yourQuery","text1;text2;text3");
```

This API call will return data and set the data into the list of control.

We can use following API to fetch data .

```
formReference.getNGDataFromDataCache( "yourQuery");
```

This API call will return list of data.

We can also use following API to fetch data on precondition.

```
formReference.getNGDataFromDataCache( "yourQuery","preCondition");
```

This API call will return data and set the data into the list of control.

78) Get Fresh Data from Table in Java

Here is the explanation of using this API with Example.

```
formReference.getDataFromDataSource( "yourQuery");
```

This API call will return a list containing data after execution of query and set the data in cache.

79) Set Data in Table in Java

Here is the explanation of using this API with Example.

```
formReference.saveDataIntoDataSource ( "yourQuery");
```

This API call will set either update or insert the data in database.

80) Set Result from Query to Controls

Here is the explanation of using this API with Example.

```
formReference.getDataFromDataSource ( "query","arrayListOfControls");
```

This API call will set fetch data from the query and put the data in the list of control in arraylist.

81) Create Zone from Java

Here is the explanation of using this API with Example.

```
formReference.NGCreateZone ("zoneabc", "100, 100, 200, 200");
```

This API call will used to create a zone , here first parameter is name of the zone and second parameter will a string of cordinates separated by comma(.). As shown in the example.

82) Map Created Zone from Control in Java

Here is the explanation of using this API with Example.

```
formReference. NGMapZones ("Text1", "zone1");
```

This API call will map the control Text1 to a existing zone **zone1**.

83) Apply Digit grouping on Text Control in Java

Here is the explanation of using this API with Example.

```
formReference. NGApplyDigitGrouping ("Text1", true);
```

This API call will set the digit grouping for the control name **Text1** .

84) Apply Digit grouping on Text Control from Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.NGApplyDigitGrouping("Text1",true);
```

This API call will set the digit grouping for the control with id **Text1** .

85) Get Whether Change fired is on FormLoad or Manual Value Change

Here is the explanation of using this API with Example.

formReference.isFormLoadChange ();

This API call will return true and false based on whether value got changed on formLoad or manually changed. If this returns false and value got changed then it means value got manually changed.

86) Get Process Instance Id in Java

Here is the explanation of using this API with Example.

formReference.getWFProcessName ();

This API call will return the name of the current process.

87) Get Process Instance Id in JS

Here is the explanation of using this API with Example.

com.newgen.omniforms.formviewer.getWFProcessName();

This API call will return the name of the current process.

88) Get Activity Name in Java

Here is the explanation of using this API with Example.

formReference.getWFActivityName ();

This API call will return the name of the Activity.

89) Get Activity Name in Js

Here is the explanation of using this API with Example.

com.newgen.omniforms.formviewer.getWFActivityName();

This API call will return the name of the **Activity**.

90) Get Cabinet Name in Java

Here is the explanation of using this API with Example.

formReference.getWFEngineName ();

This API call will return the name of the **Cabinet**.

91) Get Cabinet Name in Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.getCabinetName( );
```

This API call will return the name of the **Cabinet**.

92) Get AppServer IP in Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.getJTSIP( );
```

This API call will return the name of the **Cabinet**.

93) Get AppServer Port in Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.getJTSPort( );
```

This API call will return the Port number.

94) Set Associated Field of ListView Clearance Flag in Java

Here is the explanation of using this API with Example.

```
formReference. setM_bClearAllControls(true);
```

This API call will set the clearance flag true/false depending upon the parameter of the Activity.

95) Set Grayed Color of Disabled Field on FormLoad

Here is the explanation of using this API with Example.

```
formReference. setDisabledGray(true);
```

This API call will set the Grayed Color for Disabled field because of parameter **true**.

96) Export Form to Pdf

Here is the explanation of using this API with Example.

```
formReference.exportToPDF("formFilePDFName","C:\\Users\\Public\\Documents");
```

This API call will export the form to a file name **formFilePDFName** to above mention location i.e. To in documents folder.

If we don't pass second parameter to this API it will open a prompt window asking where to save file.

97) Print Form

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.NGPrint();
```

This API call will return print the Form.

98) Print Tab Control Sheet in js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.printNGContainer("tab1","sheet1" );
```

Above API call will print the tab sheet1 in control **tab1**.

99) Load URL in Iframe from Java

Here is the explanation of using this API with Example.

```
formReference.setIFrameSrc("iFrameName", "URLSource");
```

This API call will set the **URLSource** in the control **iFrameName**.

100) Load URL in Iframe from Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.setIFrameSrc("iFrameName", "URLSource");
```

This API call will set url equal to **URLSource** in the iframe with id **iFrameName**.

101) Save Form from Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.RaiseEvent("WFSave",true );
```

Above API call will save the form and takes the flow to next work item .

102) Save Form from Java

Here is the explanation of using this API with Example.

```
formReference.RaiseEvent("WFSave",true );
```

This API call will set the status of **radio1** button control to because of **true**.

103) Submit Form from Js

Here is the explanation of using this API with Example.

```
com.newgen.omniforms.formviewer.RaiseEvent("WFDone", true);
```

Above API call will submit the form and takes the control to next work item.

104) Set Radio button ON/OFF in Java

Here is the explanation of using this API with Example.

```
formReference.setRadioButton("radio1", true);
```

This API call will set the status of **radio1** button control to because of **true**.

105) Execute Webservice on Control in java

Here is the explanation of using this API with Example.

```
formReference.executeWebservice ("button1","onclick","serviceName","operation");
```

This API call will execute **webservice onclick** event on control **button1** on server with name **serverName** and perform the mention **operation**.

106) Fetch Fragment on Event in java

Here is the explanation of using this API with Example.

formReference.fetchFragment("frame1", "fragment1", "text1");

This API call will fetch Fragment **fragment1** into **frame1** and set value in the field **text1**.

107) AutoComplete Enable on TextField from DB Data in Java

Here is the explanation of using this API with Example.

formReference.autoComplete("text1" ,"qeuryForData",true);

This API call will set auto complete for control text1 from the queryForData from database and last flag for weather to acess cache.

Here is another API we can use.

formReference.autoComplete("text1" ,"tableName","columnName",true);

This API call will feach the data from the selected table and from a particular column .

This will show data there as sugestion and last flag for weather we want to cache the data or not.

108) Open PickList in Java

Here is the explanation of using this API with Example.

Here is the signature of the API.

```
formReference.getNGPickList(String associatedCtrl, String colHeaders, boolean  
pBatchingReq , int batchSize);
```

Here how we will be use it

```
formReference.getNGPickList("controlName", "headerfield", true,20);
```

Above API call will return a picklist object for **controlName** control and header field will be headerfield here we have enabled batching and also defined the batch size to 20.

After getting picklist object we need to call method populate

Here is the explanation of using this API with Example.

```
formReference.autoComplete("text1" ,"queryForData",true);
```

This API call will set auto complete for control text1 from the queryForData from database and last flag for weather to access cache.

Here is the explanation of using this API with Example.

```
formReference.autoComplete("text1" ,"queryForData",true);
```

This API call will set auto complete for control text1 from the queryForData from database and last flag for weather to access cache.

109) Load Data in PickList in Java

Separate document is provided for picklist Control.

110) Search Data on PickList Control in Java

Separate document is provided for picklist Control.

111) Set Value from PickList to Control in Java

Separate document is provided for picklist Control.

112) Set Control Mandatory in Java

Here is the explanation of using this API with Example.

formReference.setMandatory("text1", true);

This API call will mark a control mandatory on based upon the flag value passed in second parameter.

113) Multilingual Support through Custom Coding

User Need to create project structure as shown below:

com.newgen.omniforms.user.properties in custom coding jar.

Inside this package create a property file with name convention as :

FormName_Locale.properties

For Ex : If formname is LoanForm and locale is arabic (ar_sa).

Then create property file as LoanForm_ar_Sa.properties.

To change the labels of control fields at runtime specify Key-Value pair in properties file.
As shown below:

```
Label22 = سيبيصث  
Tab1=Sheet1;Sheet2          // SheetName semi colon separated  
Sheet1=AlternameforSheet1 // Sheet1 name to be displayed  
Sheet1=AlternameforSheet2 // Sheet2 name to be displayed
```


114) Repeater Specific API documentation

To use Repeater Specific APIs, first of all we need to fetch repeater object from Form instance like as show here;

```
FormReference formObject =FormContext.getCurrentInstance().getFormReference();  
IRepeater repeaterControl = formObject.getRepeaterControl("Frame1");
```

115) Get Total Column Count in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.getColumnCount( );
```

This API call will return total column count in Repeater.

116) Get Total Row count in a Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.getRepeaterRowCount();
```

This API call will return total row count in Repeater.

117) Get value of Control in particular row Inside Repeater.

Here is the explanation of using this API with Example.

```
repeaterControl.getValue(1,"text1");
```

This API call will return the value of Control text1 from first Row.

118) Set String value in control of a particular Row inside Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.setValue(1, "text1", "value1");
```

This API call will set value in control **text1** equal to **value1** from row with index **1**.

119) Set the List of values for Combo Box Control in particular row of Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.setValue(1, "comboBox1", listOfString);
```

This API call will set list of String from listOfString object into the **comboBox1** from a row with row index 1.

120) Sets the list of values for Combo Box in a Particular Cell in Repeater.

Here is the explanation of using this API with Example.

```
repeaterControl.setValue(1, 1, listOfString);
```

This API call will set list of String from listOfString object into the comboBox from a row with row index 1 and column index1.

121) Sets the String Value in Control in particular Row and Column in Repeater .

Here is the explanation of using this API with Example.

```
repeaterControl.setValue(1, 1, "stringValue");
```

This API call will set string value equal to **stringValue** into a particular control at a row with row **index 1** and column **index1**.

122) Whether particular Cell is Editable or Not in Repeater.

Here is the explanation of using this API with Example.

```
repeaterControl.isEditable(1, "text1");
```

This API call will return true/false based upon whether text1 control cell is editable or not in a particular row with **index 1**.

123) Set a particular control in a row editable in Repeater.

Here is the explanation of using this API with Example.

```
repeaterControl.setEditable(1, "text1",true);
```

This API call will set a particular control name text1 of a row with index1 editable based upon the third parameter pass as true/false .

124) Whether particular cell is disabled or not in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.isDisabled(1, "text1");
```

This API call will return true or false based upon whether control text1 of row 1 is disabled or enabled.

125) Set a particular cell disabled or enable in Repeater.

Here is the explanation of using this API with Example.

repeaterControl.setDisabled(1, "text1", false);

This API call will enable or disable a control on the basis of value passed to third parameter . Here in above example control text1 will be disable from row no 1.

126) Add a row in Repeater.

Here is the explanation of using this API with Example.

repeaterControl.addRow();

This API call will add a row in the repeater.

127) Whether a row is visible or not in a Repeater.

Here is the explanation of using this API with Example.

repeaterControl.isRowVisible(1);

This API call will return true/false depends upon whether row 1 is visible or not.

128) Show or Hide a Row in Repeater

Here is the explanation of using this API with Example.

repeaterControl.setRowVisible(1,false);

This API call will hide row with index 1 as we have passed false value to second parameter .

129) Show or Hide a Column in Repeater

Here is the explanation of using this API with Example.

repeaterControl.setColumnVisible(1, false);

This API call will hide column with index 1 as we have passed false value to second parameter .

130) **Clear Data of Repeater**

Here is the explanation of using this API with Example.

```
repeaterControl.clear();;
```

This API call will clear all data of repeater .

131) **Set Width of Column in Repeater.**

Here is the explanation of using this API with Example.

```
repeaterControl.setColumnWidth(1, 125);
```

This API call will set width of column with index1 to 125px .

132) **Remove a Row from Repeater.**

Here is the explanation of using this API with Example.

```
repeaterControl.removeRow(5);
```

This API call will remove row with index 5 from repeater.

133) **Get a Component for a row in Repeater**

Here is the explanation of using this API with Example.

```
repeaterControl.getComponent(5,"text1");
```

This API call will return java object representing control text1 from row with index5 .

134) **Set Style for Particular Control in a Particular Row in Repeater.**

Here is the explanation of using this API with Example.

```
repeaterControl.setStyle(1, "text1", "background-color:red;color:red;top:5");
```

This API call will set the style of text1 of a row with index1 to above mention third parameter .

135) Check whether a Control from a particular row is Visible/Invisible in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.isVisible(5, "text2");
```

This API call will return true/false depending upon the whether **text2** of a row with **index5** is visible or invisible .

136) Set Visible/Invisible a Control from a particular Row in Repeater .

Here is the explanation of using this API with Example.

```
repeaterControl.setVisible(2, "text1", false);
```

This API call will hide a control **text1** from row with Index2 as we have passed **false** flag.

137) Whether a control is Disable or Enable from a Particular Row in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.isDisabled(1, "text1");
```

This API call will return true/false depending upon the whether control **text1** is enable or disable .

138) Enable or Disable a Control from a Particular Row in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.repeaterControl.setDisabled(1,"text1", true);
```

This API call will return index of selected row .

139) Set Background-color of a Control in a Row in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.setRowBackColor(1, Color.yellow);
```

This API call will set the **background** color of a row with **index1** to **yellow**.

140) Get Repeater column Header in Repeater

Here is the explanation of using this API with Example.

```
repeaterControl.repeaterControl.getRepeaterHeaders();
```

This API call will return a list of String objects each representing header of a column.

141) **Set Repeater column Header in Repeater**

Here is the explanation of using this API with Example.

repeaterControl. repeaterControl.setRepeaterHeaders(listOfString);

This API call will set the **headers** of column from the **listOfString**.
Here listOfString is arrayList of String .

142) **Set Color of Add and Delete Row link in Repeater.**

Here is the explanation of using this API with Example.

repeaterControl.setRepeaterLinkColor(Color.yellow);

This API call will set the color of add and remove icon to yellow .

143) Get Data from Stored Procedure

Following is the signature of the of the API method

```
getDataFromStoredProcedure(String pProcName, List pParamList);
```

Here a call to above API explaining the usage .

```
formReference.getDataFromStoredProcedure("your store procedure name",  
                                         listOfParameter);
```

In above API call we will pass first parameter store procedure name as string and in second parameter we will pass a arrayList object containing the parameters for the stored procedure.

For example :

For example , if procedure expects two parameters of type Integer, Text .
Parameterlist will contain datatype and value both .DataType and value will be colon (:) separated .
In this case paramlist would be like as follows:

```
List<String> paramlist =new ArrayList<String>( );  
paramlist . add ("Integer : 10");  
paramlist . add ("Text : John");
```

Note:

- 1) **In above example we have seen the second parameter to API is an arrayList object. This arrayList object will have a list of parameters each of those parameter will be colon(:) separated from its Data Type as show above while preparing arrayList object.**
- 2) **In case, procedure expects no parameter , pParamList will be empty.**
- 3) **supported parameter dataTypes are (Integer,Text,Float) .**

144) CustomCoding Regarding PickList GuideLines

To perform operation on pick list we need to get picklist object as shown here :

```
FormReference formObject = FormContext.getCurrentInstance().getFormReference();  
PickList objPickList =formObject.getNGPickList("Text1","UserName",true,3);
```

After this we will use **objPickList** for all the operation on the pick list.

145) Populate PickList With Database query results

Here is the explanation of using this API with Example.

```
objPickList.getRecordFetchedInBatch();
```

This API call will return the no of records return .

146) Get No Of Records Fetched in Batch

Here is the explanation of using this API with Example.

```
objPickList.getRecordFetchedInBatch();
```

This API call will return the no of record fetched .

147) To Enable or Disable Next or Previous Button in PickList.

Here is the explanation of using this API with Example.

```
objPickList.enableButton("Next", true);  
objPickList.enableButton("Previous", true);
```

This API call will set the Next and Previous button as we have pass true flag for both .

148) To Get the filter value typed in searched Text Box in PickList

Here is the explanation of using this API with Example.

```
objPickList.getSearchFilterValue();
```

Above API call will return the text value in filter text box.

149) Get The Value of PickList from specified Row and Column Index

Here is the explanation of using this API with Example.

objPickList.getValueAt(1, 1);

This API call will return the value of the cell with rowIndex1 and colIndex1.

150) Get the Total Records of PickList

Here is the explanation of using this API with Example.

objPickList.getM_iTotalRecordsFetched();

This API call will return the total Records in a picklist.

151) Get the Selected Row Value from a PickList

Here is the explanation of using this API with Example.

objPickList.getSelectedValue();

This API call will return a list of String values of selected rows in a picklist.

152) Attach Custom Event Listener With PickList

Write Down EventListenerHandler Class which extends EventListenerImplementor

as shown below. Here you can specify custom coding event which will be executed on click of **OK,CANCEL,NEXT,PREVIOUS** button of picklist.

```
package com.newgen.omniforms.user;

import com.newgen.omniforms.component.Form;
import com.newgen.omniforms.component.PickList;
import com.newgen.omniforms.component.TextBox;
import com.newgen.omniforms.component.behavior.EventListenerImplementor;
import com.newgen.omniforms.context.FormContext;
import com.newgen.omniforms.model.NGListDataModel;
import com.newgen.omniforms.util.Constant.EVENT;
import com.newgen.omniforms.util.OFUtility;
import java.util.List;
```

```

import javax.faces.event.ActionEvent;

public class EventListenerHandler extends EventListenerImplementor{

    public EventListenerHandler( String picklistid )

    {
        super(picklistid);
    }


    public EventListenerHandler(String picklistid , EVENT compId){

        super(picklistid,compId);

    }

    @Override

    public void btnNext_Clicked(ActionEvent ae) {

        //PickList objPckList =
        FormContext.getCurrentInstance().getFormReference().getNGPickList(true);

        //System.out.println(" Fetched Records = " +
        objPckList.getM_iTotalRecordsFetched());

    }


    @Override

    public void btnSearch_Clicked(ActionEvent ae){

        System.out.println("Inside method btnSearch_Clicked");

        PickList m_objPickList = FormContext.getCurrentInstance().getDefaultPickList();

        String filter_value=m_objPickList.getSearchFilterValue();

        System.out.println("Filter value : " + filter_value) ;

        m_objPickList.setBatchRequired(true);

        m_objPickList.setBatchSize(10);

        System.out.println("m_objPickList.getM_iTotalRecordsFetched() : " +
        m_objPickList.getM_iTotalRecordsFetched()) ;

        //m_objPickList =
        FormContext.getCurrentInstance().getDefaultPickList();
    }

```

```

        m_objPickList.setVisible(true);
    }

    @Override
    public void btnOk_Clicked(ActionEvent ae) {

        PickList m_objPickList =
        FormContext.getCurrentInstance().getDefaultPickList();

        Form obj = ( Form)FormContext.getCurrentInstance().getFormReference();

        TextBox comp = ( TextBox ) obj.getComponent("Text1");

        comp.setValue(m_objPickList.getSelectedValue().get(0));

        OFUtility.render(comp);
    }

    @Override
    public void btnPrev_Clicked(ActionEvent ae) {

        // Write Custom Code here

    }
}

```

Note :

If there are Two or more PickList you can differentiate between its action using
`m_objPickList.getAssociatedTxtCtrl()` // It will specify which picklist is current in open state.

Usage :

```

PickList objPickList = formObject.getNGPickList("Text1" , "MyColumns1" , true , 3);
objPickList.addPickListListener(newEventListenerHandler(objPickList.getClientId()));

```

EventListenerHandler is the name of the custom class as shown above.

153) Process Specific CSS option:

CSS can be defined following ways:

Scenario:

1. If user wants the CSS to be generic for all processes and forms deployed on a server then place customCSS.css at location
webdesktop.war/resources/en_us/css/customCSS.css

2. If user wants the CSS to be process and form specific then place the define CSS with form name and place inside the process folder as
webdesktop.war/resources/en_us/css/processname/formname_CSS.css

ex:

webdesktop.war/resources/en_us/css/prc1/Form1_CSS.css

webdesktop.war/resources/en_us/css/prc1/Form2_CSS.css

webdesktop.war/resources/en_us/css/prc2/Form1_CSS.css

1. Provided setCSSClass custom method :

User can override the css of controls **Button**, **Label**, **TextBox** and **Datepicker** using **setCSSClass("controlld", "cssClass");**

For ex:

```
formObject.setCSSClass("Button2", "Class4");
```

2. Custom CSS for Collapsible Frame header/ Repeater:

User can write its own css to override the Frame's header or Repeater's header css.

While writing the custom css, must follow the format.

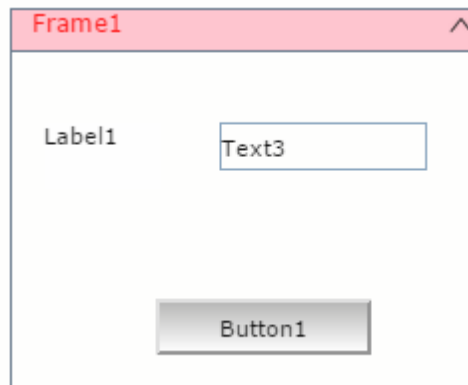
Format: **Frameld_Header_CSS{**

CSS Code goes here.

}

Ex:

```
.Frame1_Header_CSS{  
    color: red !important ;  
    background-color: pink !important ;  
    FONT-SIZE: 12px ;  
}
```



3. CSS Support for Repeater's Header, if Header fields are added through custom code.

If user uses `setRepeaterHeaders()` API to add repeater's header and wants to write its own css for Repeater's header, he needs to define css class in his custom css file.

For ex:

Suppose he is adding header in Frame2 control through custom code like below:

```
IRepeater rep = formObject.getRepeaterControl("Frame2");  
String headers = "Name,Age";  
List<String> listHeader = new ArrayList<String>();  
listHeader.addAll(Arrays.asList(headers.split(",")));  
rep.setRepeaterHeaders(listHeader);
```

Format: **FrameId_Header_Table{ }**

Ex:

```
.Frame2_Header_Table{  
    background-color: pink !important;  
    color: red !important ;  
}
```

Frame2	
Name	Age