SAP NetWeaver Business Process Management: Flash Island Meets BPM

Applies to:

SAP enhancement package 1 for SAP NetWeaver NetWeaver Composition Environment (CE) 7.1 and above.

Summary

This article will demonstrate how to make use of Flash Islands in order to integrate Adobe Flex based user interfaces into processes of SAP NetWeaver Business Process Management (BPM).

Author: Martin Möller

Company: SAP AG

Created on: 09 April 2009

Author Bio

Martin joined SAP in 2003 for his studies of applied computer science at the *University of Cooperative Education* ("Berufsakademie") in Karlsruhe. After graduating in 2006 he continued working at SAP in the business process management related area of SAP NetWeaver. Today Martin is part of the development team building SAP NetWeaver Business Process Management ("Galaxy").

Table of Contents

Introduction

SAP NetWeaver Business Process Management (BPM) allows using Web Dynpro Java user interfaces whenever a so called *human activity* in the process needs to be executed. As soon as this particular part of the process is reached the user will receive a task in the Universal Worklist (UWL). Opening it will bring up the embedded user interfaces and allows the user to interact within the process.

This little sample will demonstrate what steps need to be taken to use Adobe Flex based user interfaces in SAP NetWeaver BPM. It comprises data flow from the process context via Web Dynpro to the Flex application as well as firing events in order to continue the process flow.

Creating the Flex Component

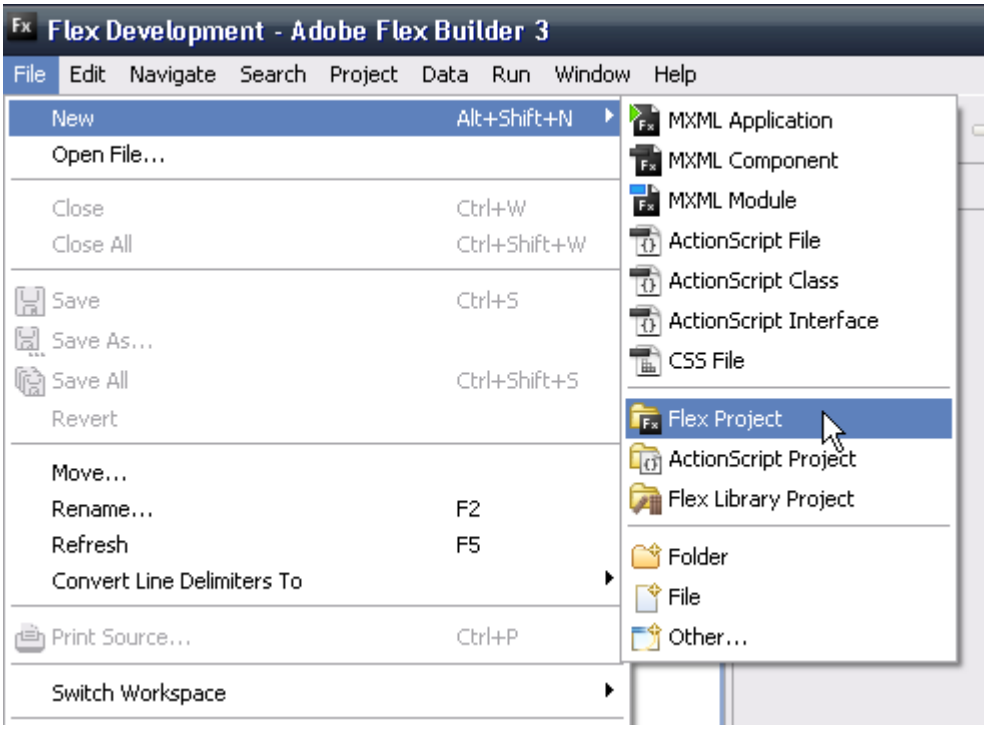
Before you could start with the Flex part you need the following:

- Adobe Flex Builder (2 or above)
- Adobe Flash Player (9 Update 3 or above)
- Flash Island Flex library **`WDIslandLibrary.swc`**.

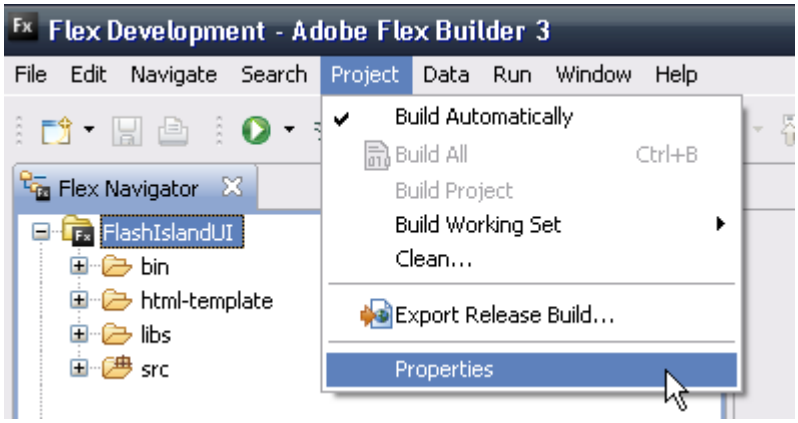
The library is responsible for the communication between Flex and the Web Dynpro Framework. It is available under the following path of the SAP NetWeaver Composition Environment Application Server:

```
\usr\sap\<SID>\J<instance_number>\j2ee\cluster\apps\sap.com\tc~wd~dispwda\servlet_jsp
\webdynpro\resources\sap.com\tc~wd~dispwda\root\global\activeComp\
```

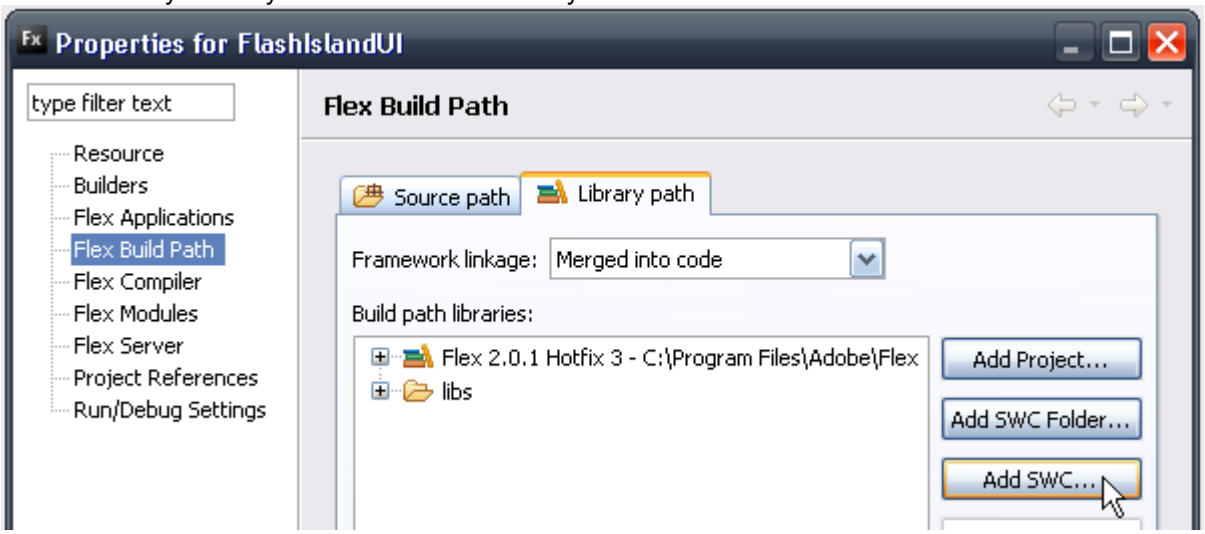
1. Create a new project, e.g. called *FlashIslandUI*.



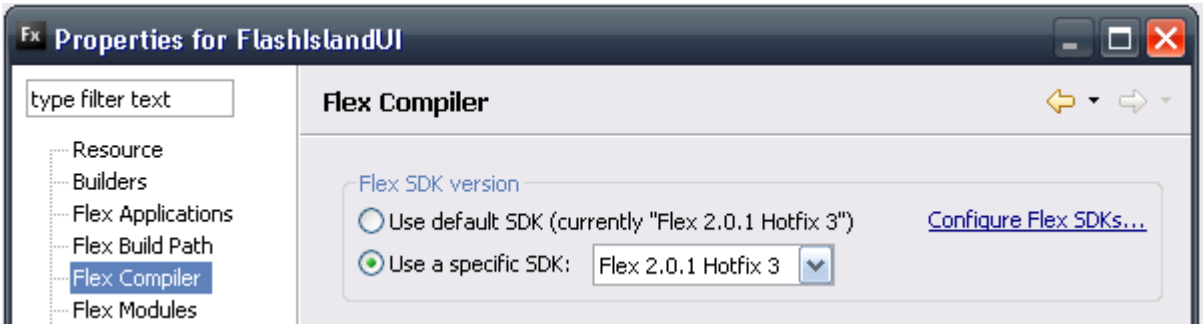
2. Open the properties of the project.



3. Add **`WDIslandLibrary.swc`** to the library path of the project by choosing *Add SWC...* Browse to the directory where you have stored the library and add it.



4. Switch the Flex Compiler to *Flex 2.0.1. Hotfix 3*



5. Now you're ready to start.

Modeling the User Interface

Open the MXML editor of your application (e.g. *FlashIsland.mxml*) and model the following UI using the design editor:

Adding the Programming Logic

Now switch to the source editor of your application and modify it accordingly:

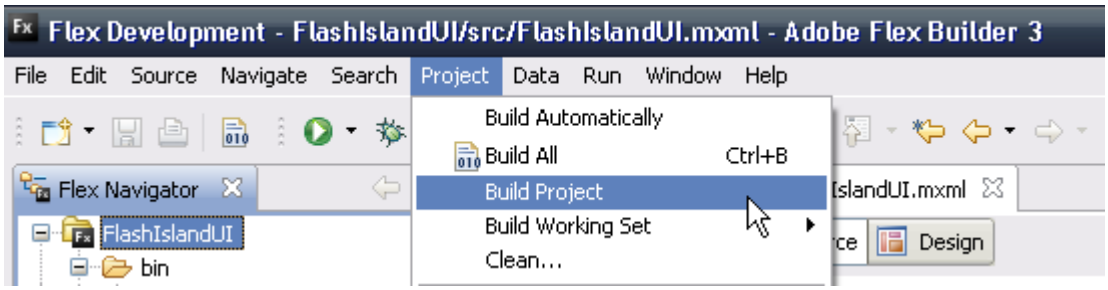
```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
      layout="absolute" creationComplete="init();">

      <mx:Script>
            <![CDATA[
                  import sap.FlashIsland;

                  [Bindable]
                  public var myProperty:Object;

                  private function init():void {
                        FlashIsland.register(this);
                  }

                  private function complete():void {
                        myProperty = myTextInput.text;
                        FlashIsland.fireEvent(this, "flexComplete", null);
                  }
            ]]>
      </mx:Script>

      <mx:TitleWindow x="10" y="10" width="320" height="110"
            layout="absolute" title="Hello World!">
            <mx:Label x="10" y="12" text="Your Name:"/>
            <mx:TextInput x="85" y="10" text="{myProperty}" width="200"
                  id="myTextInput" />
            <mx:Button x="10" y="38" label="Submit" click="complete()"/>
      </mx:TitleWindow>

</mx:Application>
```

Note the following highlights:

- After the application is created the *creationComplete* event is triggered. This will cause calling the function *init()* which subsequently will register the application at the Flash Island.
- There is a variable called *myProperty*. This variable will be used for the communication between Flex and Web Dynpro. By tagging it as bindable changes to the variable will automatically be propagated using generated events. Flash Island can react onto it and also inform Web Dynpro about the change. The variable is also used as value for the text input.
- The *complete()* function is registered as event handler for the *click* event of the button. Once this function is called it changes the value of *myProperty* and fires an event to the Flash Island. This will then be communicated to Web Dynpro which could react on it.

Build the SWF File

For the next steps we need the generated SWF file, e.g. *FlashIslandUI.swf*. After a successful build it should be located in the output folder, e.g. *bin-debug*. Typically Flex Builder works with automatic build enabled. Thus whenever you change and save your application a rebuild is triggered automatically. If that is not the case you could still trigger the build manually:

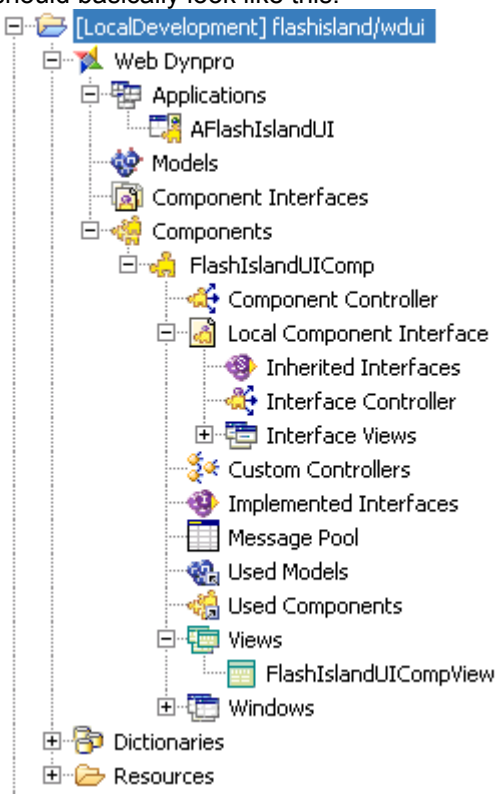


Note: This option is only enabled in case “Build Automatically” is not checked.

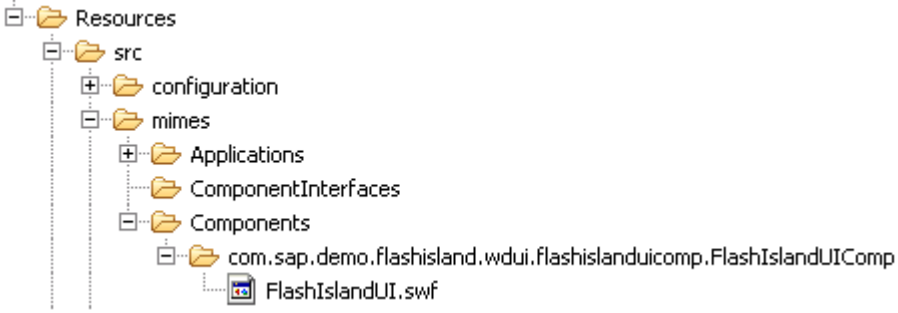
Web Dynpro, Flex & SAP NetWeaver BPM

You now have finished all steps needed on Flex site for creating a Flash Island that can be used in Web Dynpro Java. Let's continue with the Web Dynpro part:

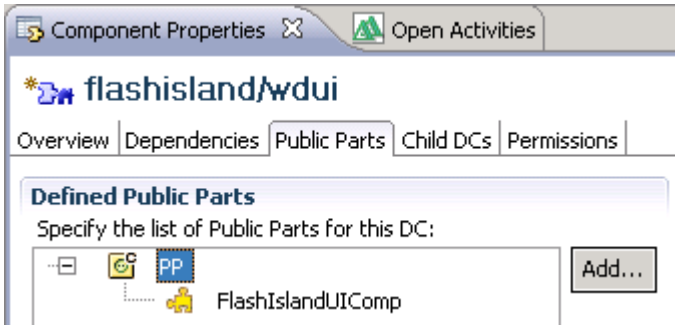
1. Create a Web Dynpro Development Component (DC), e.g. *flashisland/wdui*. The project structure should basically look like this:



2. Import the *FlashIslandUI.swf* file into the mime repository (<*Web Dynpro DC name*> → *Resources* → *src* → *mimes* → *Components* → <*component name*>):



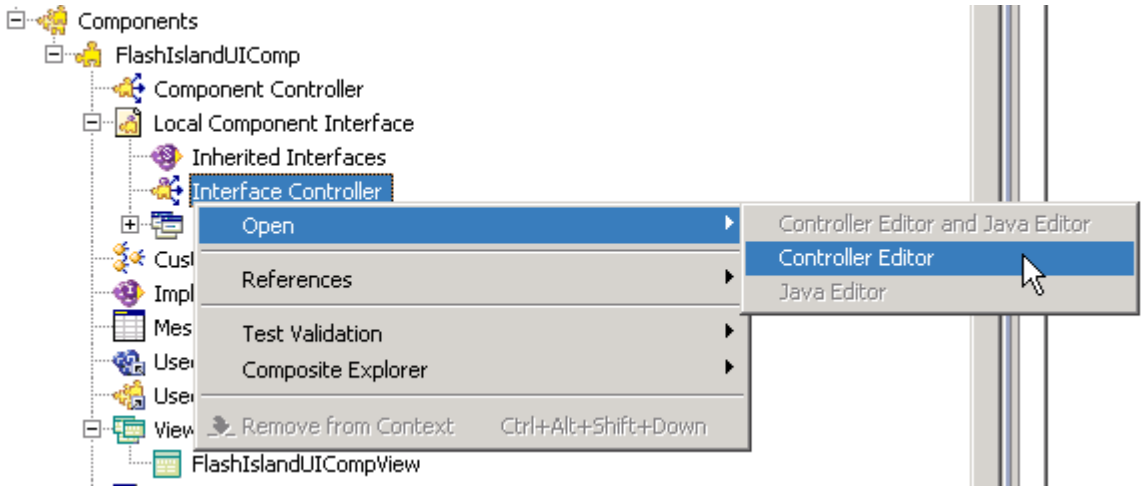
3. Create a *Public Part* so that SAP NetWeaver BPM can make use of your Web Dynpro component.
   a. Open the *Components Properties* view (*Window* → *Show View* → *Others* → *Components Properties*)
   b. Switch to the *Public Parts* tab in that view
   c. Select *Add…* and add a new public part for *compilation*.
   d. Open *Manage Entities* via the context menu of the new public part.
   e. Finally add your component to it.



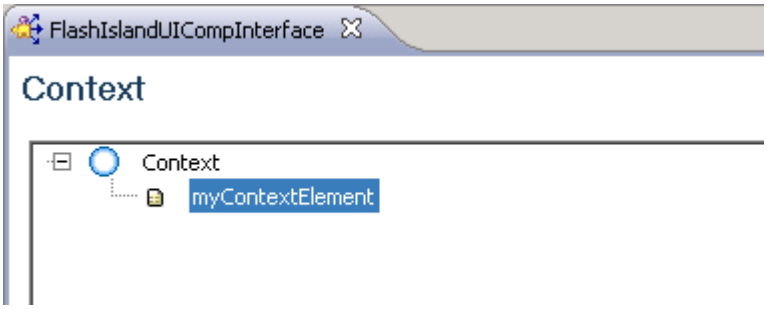
Web Dynpro & SAP NetWeaver BPM (Preparing the Interface Controller & Component Controller)

In order to establish a communication between Web Dynpro and SAP NetWeaver BPM the data context needs to be specified in the *Interface Controller*. Thereby data flows into the component when initializing it and data will be retrieved from the component once the process flow continues. In order to decide when the latter should happen SAP NetWeaver BPM listens to an event which is specified in the *Interface Controller*. Thus both – the data structure and the event – need to be created first:

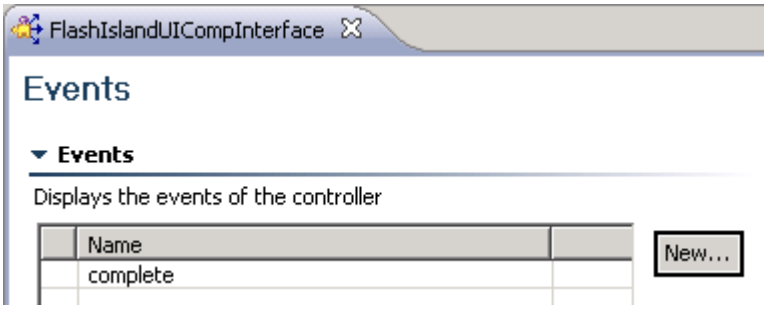
1. Open the *Interface Controller* editor



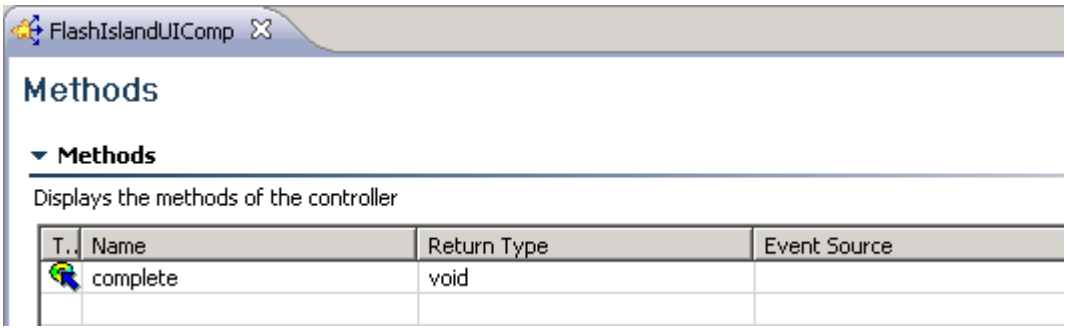
2. Go to the *Context* tab and add a single context parameter named *myContextElement* of type *String*



3. Switch to the *Events* tab and add a new event called *complete* without parameters.



4. Apply those changes to the *Component Controller* to fulfill the API of the *Interface Controller*.
5. Switch to the *Methods* tab and add *complete* method to the *Component Controller*.



This should fire the *complete* event of the *Component Controller*. Thus switch to the *Java editor*, navigate to the methods Web Dynpro already generated and add the following code snippet:

```
wdThis.wdFireEventComplete();
```

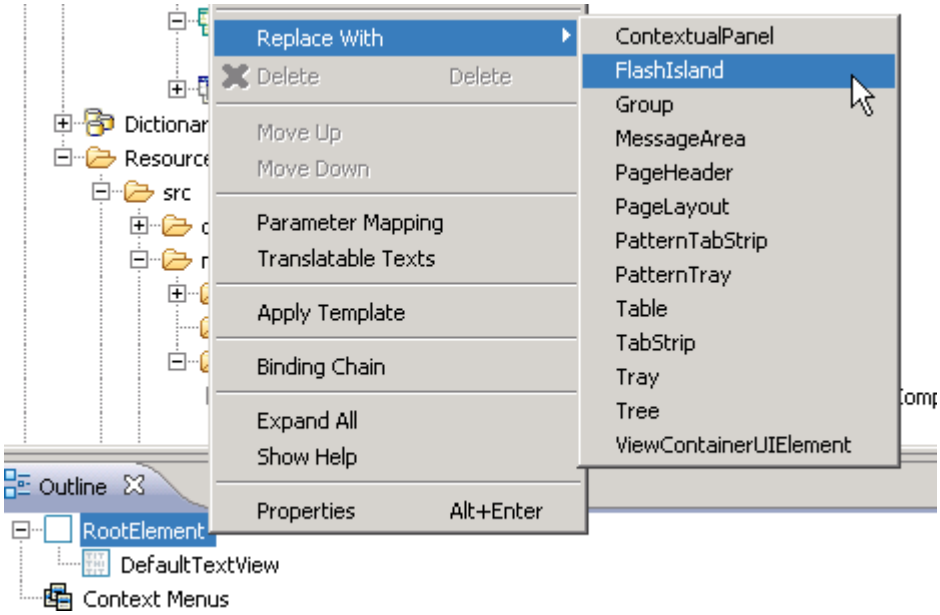
Now the view is able to trigger the event by calling this particular method.

6. Finally map the Web Dynpro context of the *Component Controller* to the *FlashIslandUICompView* view. Therefore open the *view editor* and switch to the *Context* tab. Create a new attribute and select *Create With Mapping*. Add the required *Component Controller* and choose the *myContextElement* from the *Component Controller* attribute context.

Web Dynpro & Flex (Preparing the View)

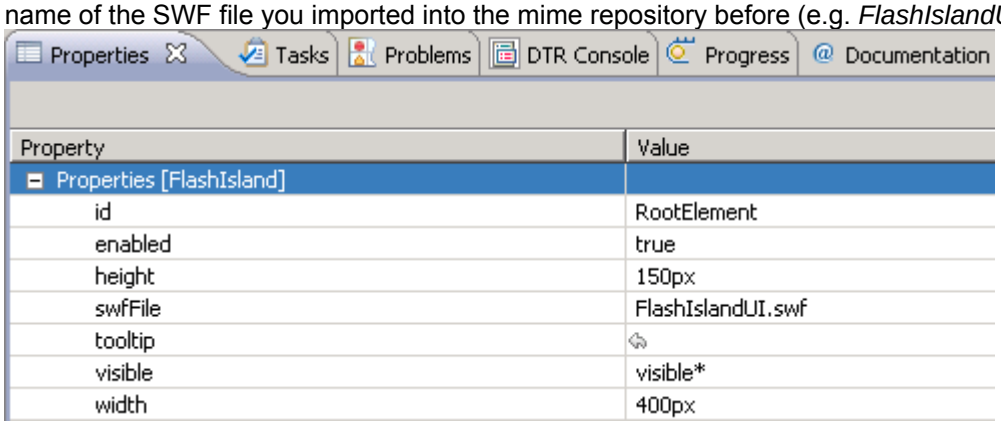
The Flash Island UI element spans an entire view. That's why you have to replace its root element.

1. Open the *FlashIslandUICompView* view and replace the *RootElement's* default container with a *FlashIsland*. To do so, select the *RootElement*, open the context menu and choose *Replace With... → FlashIsland*.



Note: Keep in mind that 'Replace With > FlashIsland' will make the new element to the new root element of the view. All elements that were children of the former root element will then be deleted.

2. Specify appropriate *width* and *height* values to avoid scroll bars to appear. Furthermore specify the name of the SWF file you imported into the mime repository before (e.g. *FlashIslandUI.swf*)



Note: The values for *width* and *height* need to be specified in pixel. The space where it will be embedded is flexible and has no specific width or height. Thus using relative values would result in a dimension of 0x0 pixel and the Flash Island would not be visible at all.

3. Insert the following elements below the new *RootElement* and adapt their properties as follows:

   - *Property* (`GACProperty`)
     - id = myProperty
     - name = myProperty

- readOnly = false
- value = myContextElement (from the view context)

o *Event* (`GACEvent`)

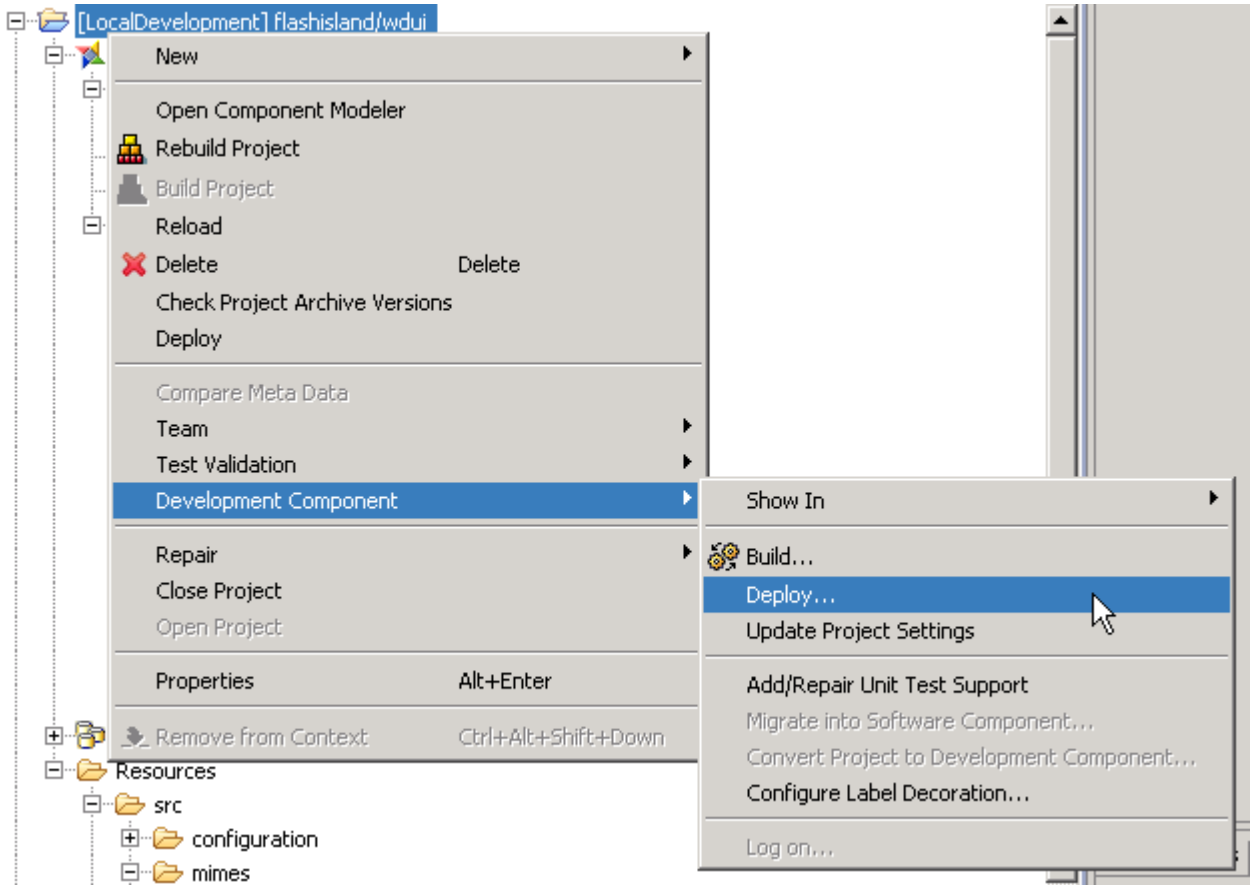
- id = flexComplete
- name = flexComplete
- onAction = *<create a new one, e.g SubmittedViaFlex >*

4. Open the *Java editor* of the view to implement *onActionSubmittedViaFlex.* Add the following code:

```
wdThis.wdGetFlashIslandUICompController().complete();
```

Build & Deploy

Great work! Now your user interface based on Flash Island is ready to be used within any SAP NetWeaver BPM process. Build & deploy your DC to make it available on the application server. Right-click on your project, select *Development Component* and then *Deploy...*

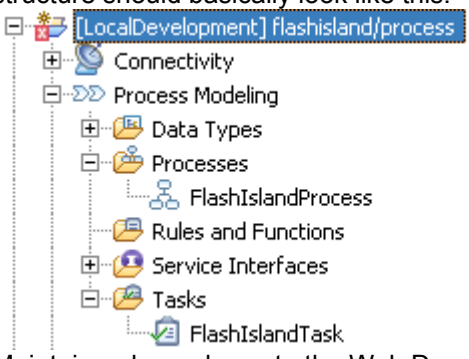


Note: Typically the SAP NetWeaver Developer Studio performs an automatic build in case the sources were modified. If not you could trigger a manual build via the same way.

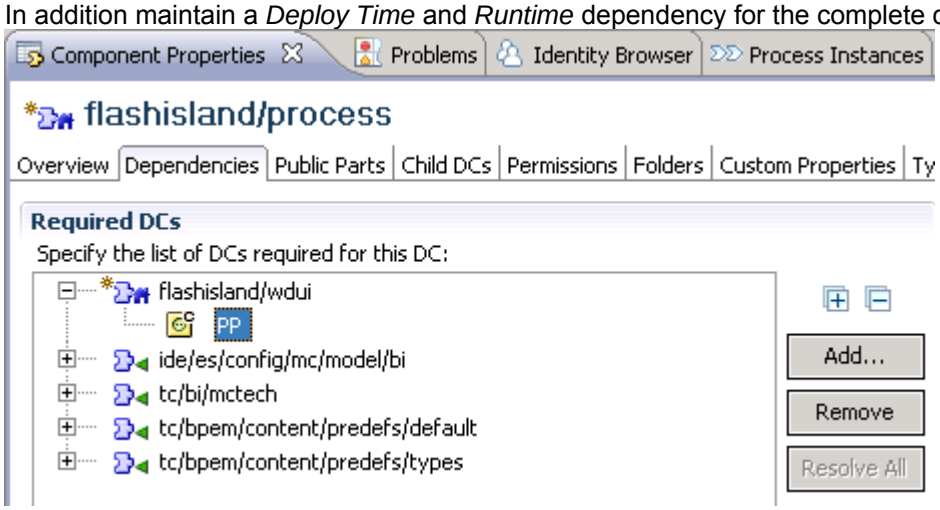
Optional: Modeling a Basic Demo Process

Now you're completely ready with the UI part and could use the newly created Web Dynpro component within your process model. If you do not have one at hand right now take a look at the following steps to create a new one from scratch.

1. Create a Process Composer Development Component (DC), e.g. *flashisland/process*. The project structure should basically look like this:

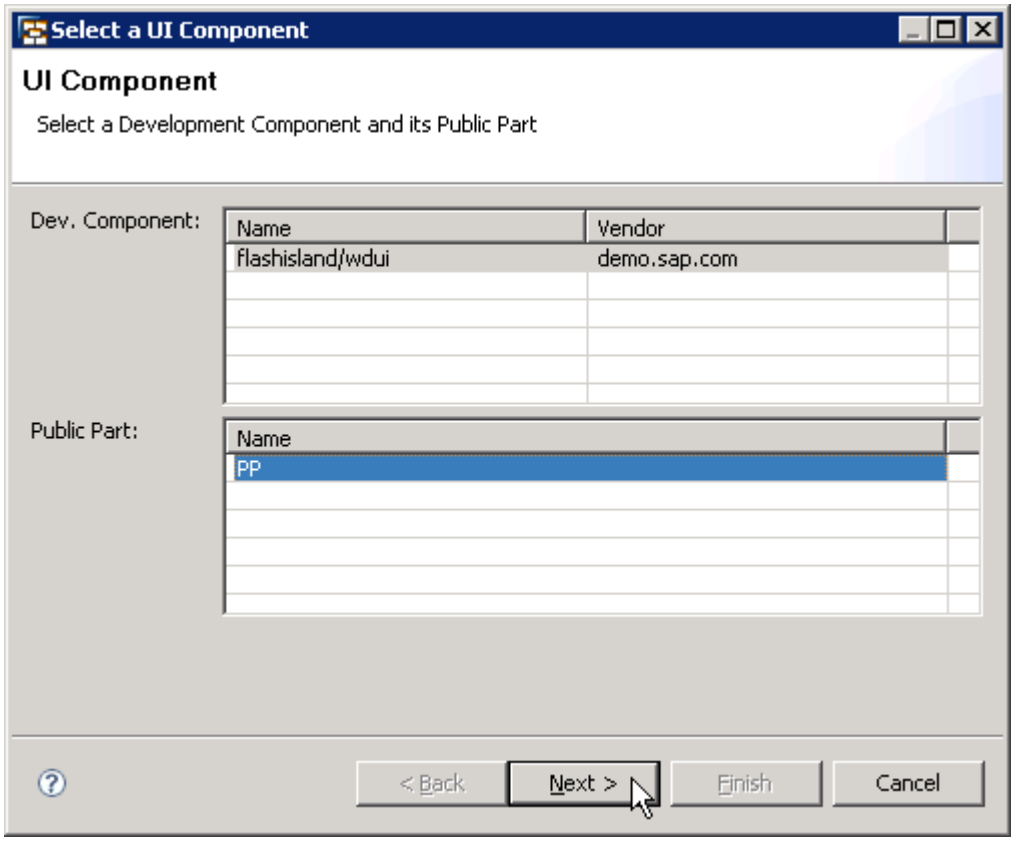


2. Maintain a dependency to the Web Dynpro project *flashisland/*wdui so that SAP NetWeaver BPM can make use of the contained user interface components.
   a. Open the *Components Properties* view (*Window → Show View → Others → Components Properties*)
   b. Switch to the *Dependencies* tab in that view
   c. Select *Add…* and search for the DC (*flashisland/wdui*).
   d. Select a *Build Time* dependency for the *Public Part.*
   e. In addition maintain a *Deploy Time* and *Runtime* dependency for the complete component.

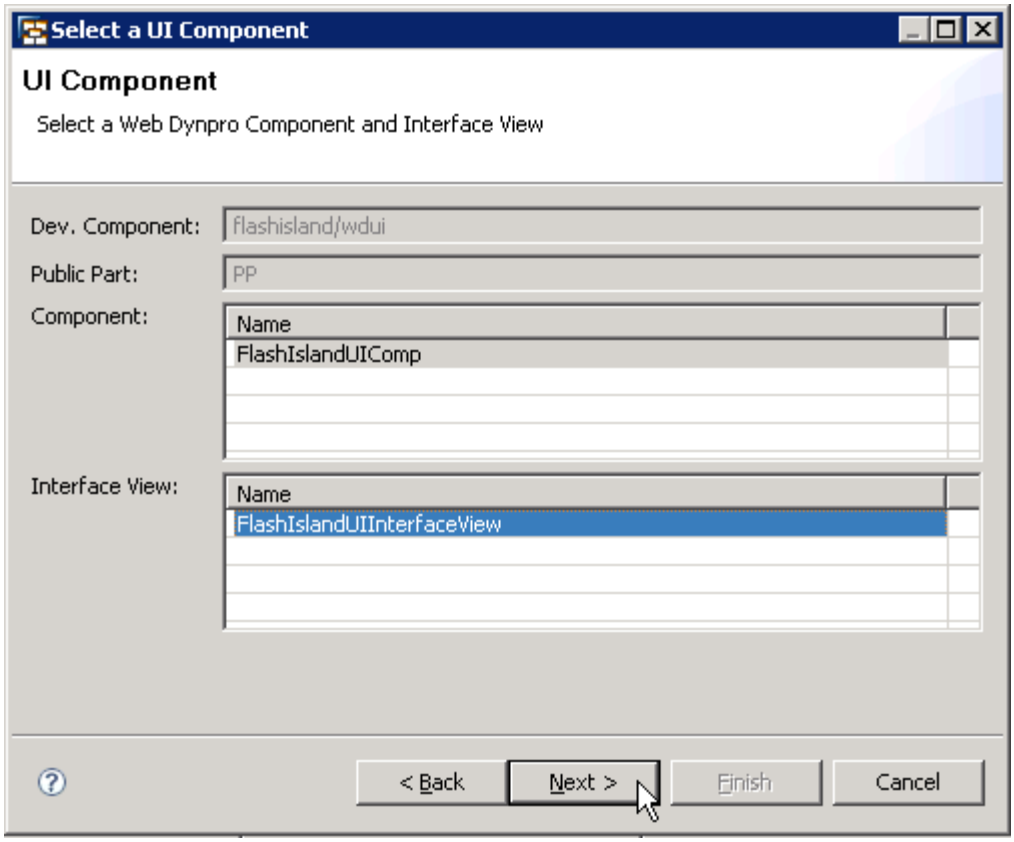


Defining the Task

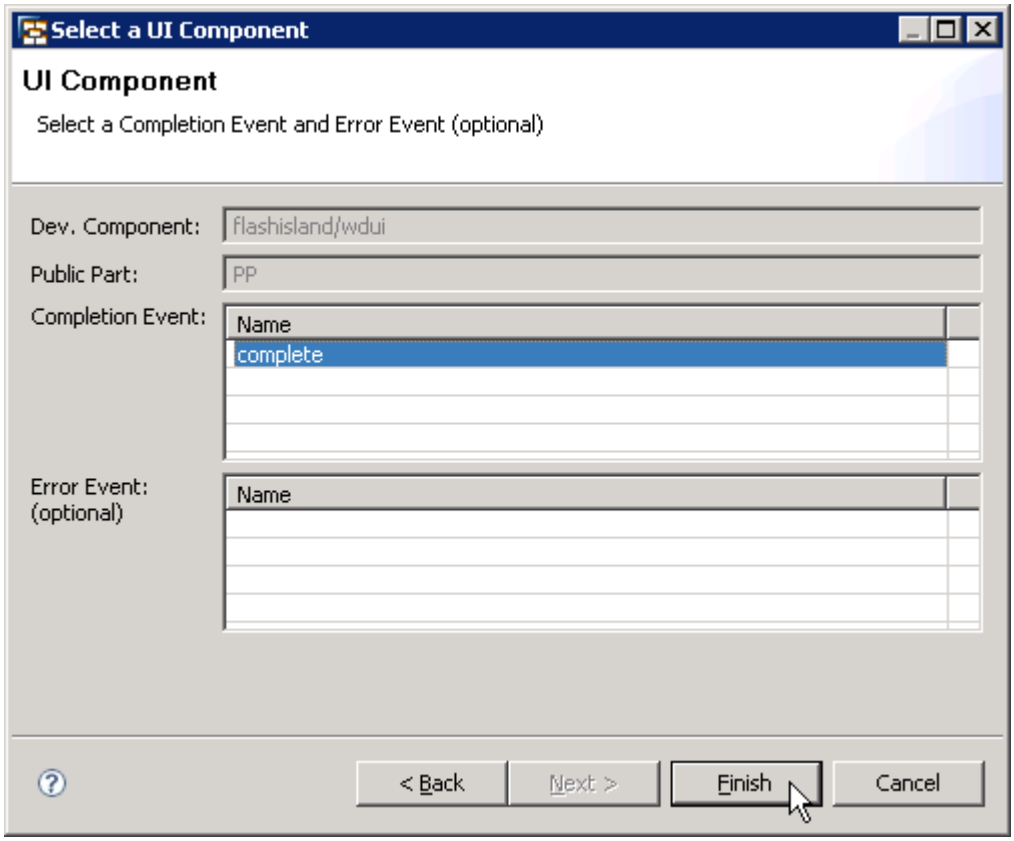
1. Open the *task editor* for the *FlashIslandTask* task. Switch to the *Overview* tab. In the *User Interface* section select the *Choose…* button. A wizard shows up which helps you selecting the right user interface. First pick the correct DC (*flashisland/wdui*) and the appropriate *Public Part* (*PP*).



In the next step, choose the right component and the interface view

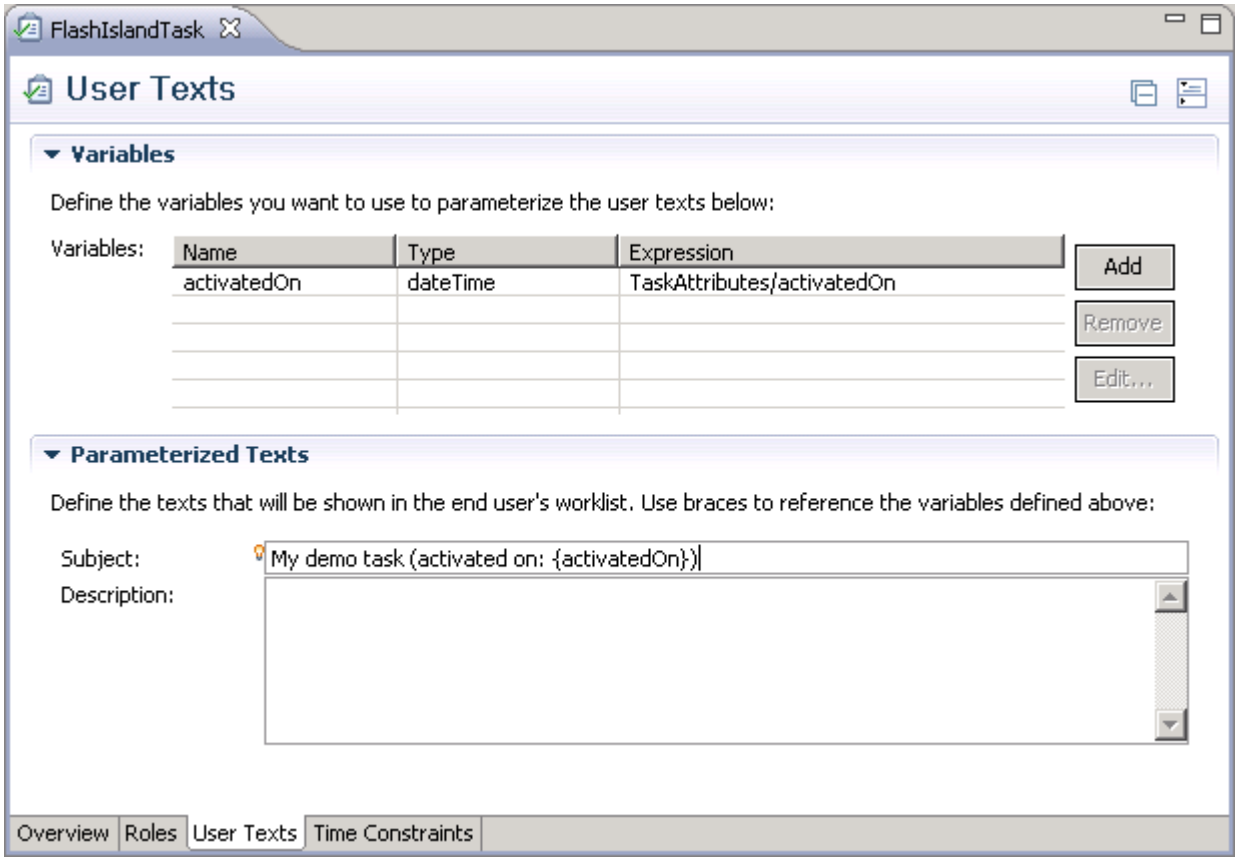


Finally select an event that indicates the completion of the task.



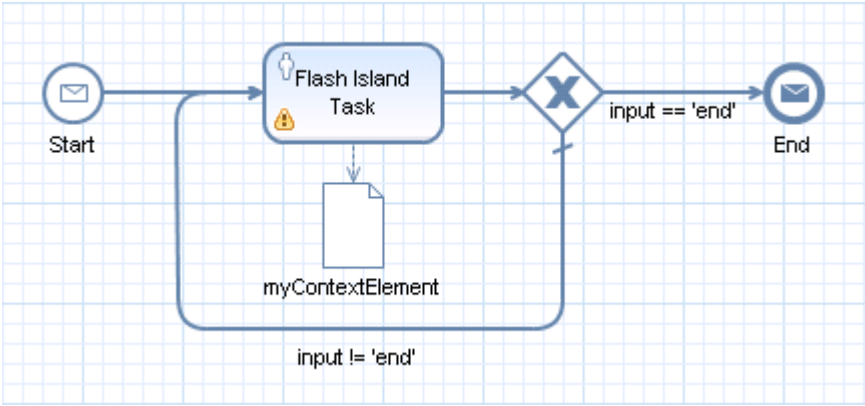
Complete the user interface assignment by selecting *Finish.*

2. Switch to the *Roles* tab and assign potential owners and administrators to this task. This setting depends on the users available in the User Management Engine (UME) on your application server. For demo purposes you could use the one single user for both.
3. Finally open the *User Texts* tab to specify a more specific subject for instances of this task. The example in the screenshot uses a dynamic parameter. Thereby the date the task was activated will be generated into the subject and eases the distinction between several task instances of the same definition.



Modeling the Process

1. Open the *process editor* of the *FlashIslandProcess* process. Model the following:



2. In order to assign the recently created *FlashIslandTask* task simply drag it onto the *Human Activity.*
3. The data object *myContextElement* is of type s*tring (http://www.w3.org/2001/XMLSchema)*.
4. Perform the following mappings in the *properties* view of each element:

   Start Event (Output Mapping)

   "<Enter some value>" (Literal) to *myContextElement (DataObject)*

   Flash Island Task (Input Mapping)

   myContextElement (DataObject) to *TaskInput.UIRequest.myContextElement*

   Flash Island Task (Output Mapping)

   TaskOutput.UIResponse.myContextElement to *myContextElement (DataObject)*

5. Assign the following conditions to the *Exclusive Choice* gateway:
   a. *Input == 'end'* *string-equal(myContextElement, "end")*
   b. *Input != 'end'* *Default Gate*
6. Click onto the checkered canvas and open its *properties* view to assign an administrator and to specify a reasonable subject as you did before with the task.

Build & Deploy

Finally build & deploy the process model to your SAP NetWeaver CE application server. Right-click onto the project → *Development Component* → *Deploy*.

Note: Typically the SAP NetWeaver Developer Studio performs an automatic build in case the sources were modified. If not you could trigger a manual build via the same way.

Running the Sample

Prerequisite for running this sample is a ready to use installation of SAP NetWeaver CE including SAP NetWeaver Business Process Management.

Starting the Process

Once all components are deployed correctly start off a new instance of the new process model

1. Log on to the SAP NetWeaver Administrator (http://host:port/nwa) as an administrator and navigate to the *Process Repository*.(*Configuration Management → Processes & Tasks → Process Repository*)
2. Select the component which contains the process model e.g.*flashisland/process*.
3. Then choose the active version of the component.
4. Finally mark the process definition (*FlashIslandProcess*) and click *Start Process.* A new UI – the so called p*rocess starter UI* – will pop up.
5. In principle you could define additional start parameters in the *process starter UI*. As the sample process does not contain any additional parameters simply choose *Start Process*.

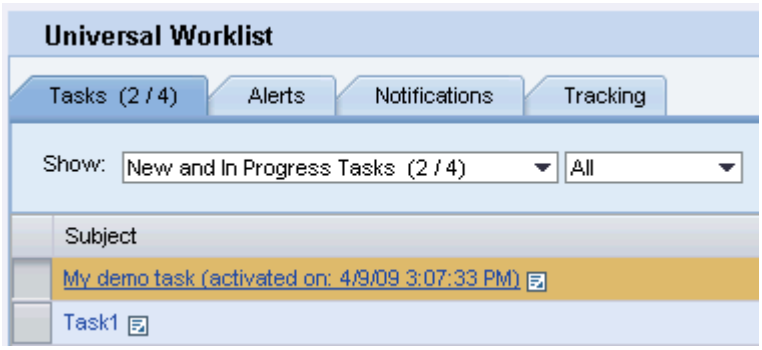
   A successful process start is indicated by a message containing the ID of the newly created process:

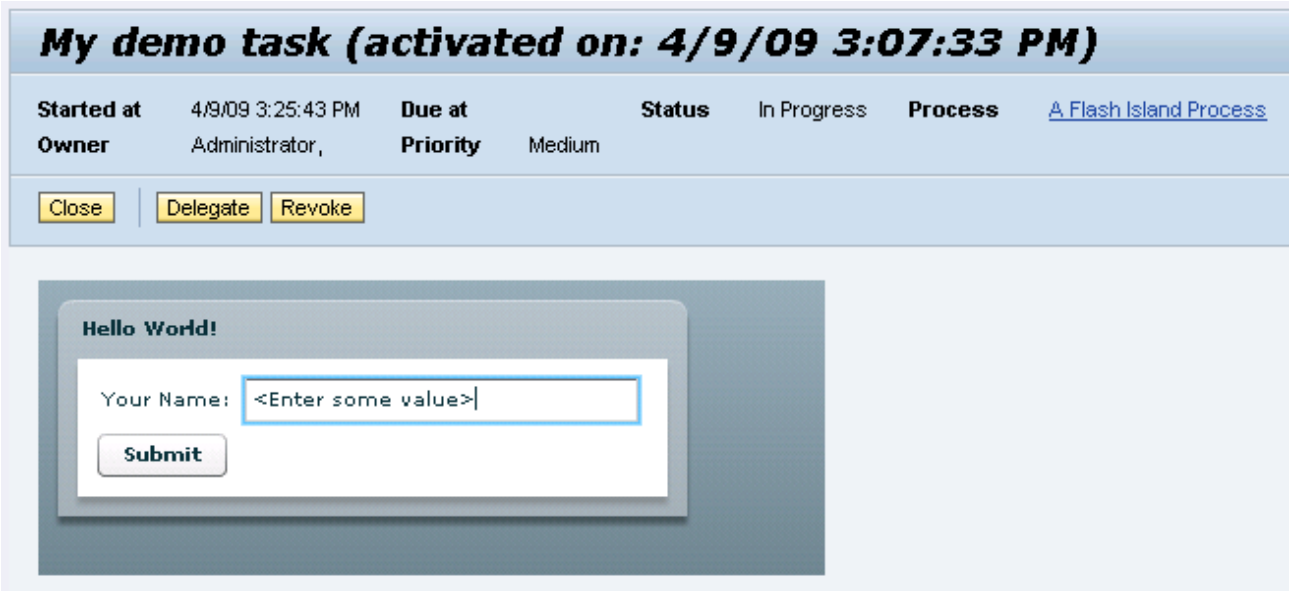
Opening the Task

Now a process instance has been started. The first task (*Flash Island Task*) will be created and gets visible in the Universal Worklist (UWL) of each *potential owner*.

1. Log on to the Portal (http://host:port/irj) as one of the potential owners that were defined when modeling the process.
2. Navigate to your UWL.
3. A new task should have appeared in the inbox. If the list did not update automatically perform a manual refresh.

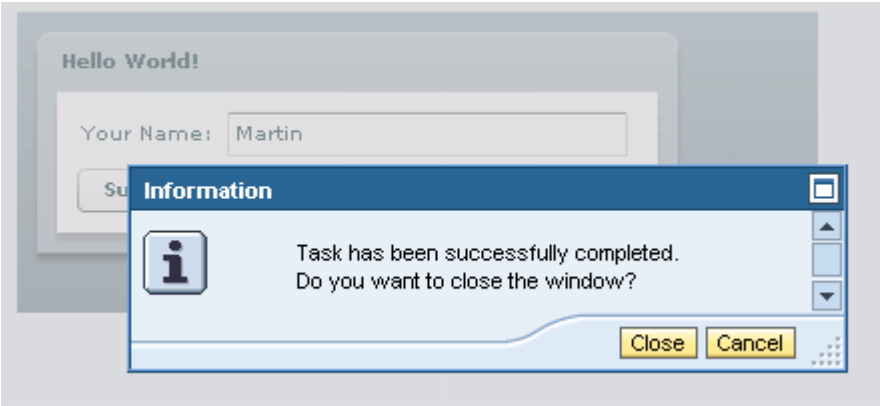
   

4. By clicking onto the task a new window should open up – the *task execution UI*. This default UI shipped with SAP NetWeaver BPM contains additional information about a task instance. Furthermore you have the possibility to add notes and attachments. Last, but not least it also embeds the user interface that was specified during design time. If everything worked out correctly you should now be able to see your Flex based user interface.

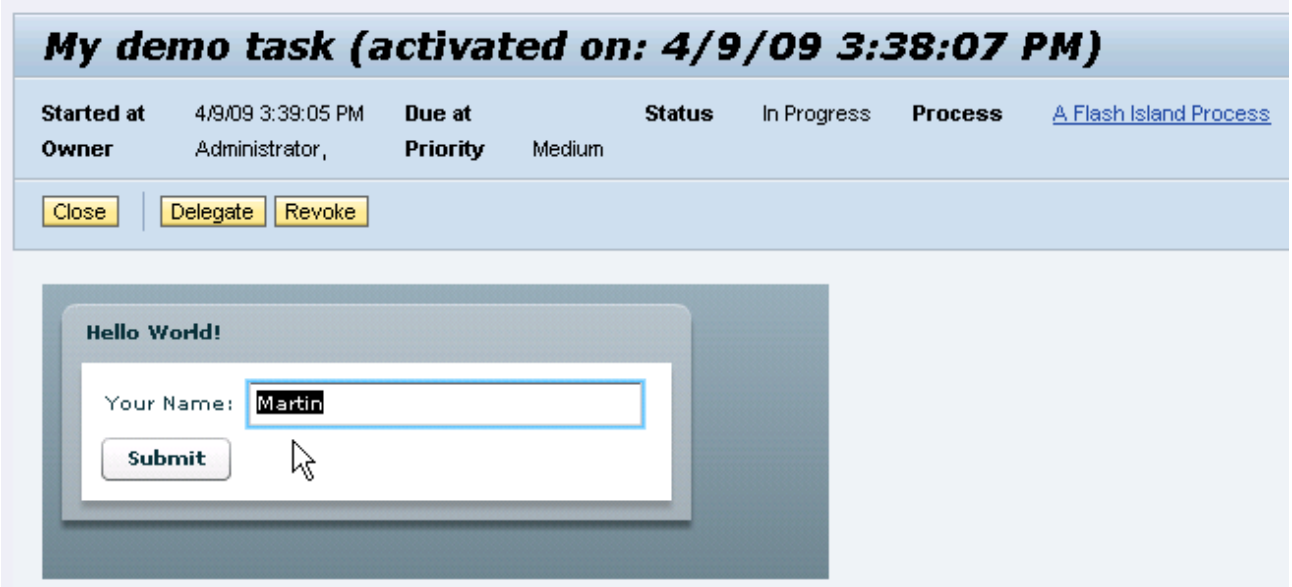


Congratulations! With the power of Flash Islands for Web Dynpro you finally created your first SAP NetWeaver BPM process containing an Adobe Flex based user interface.

Enter some data (e.g. your name) and choose *Submit*. The task should be completed.



After closing the window and refreshing the UWL it should not be part of your worklist anymore. In exchange a new task has been created (have a look at the date in the subject to see the difference). Note that the data mapping also worked seamlessly and the name you entered before is now shown in the Flex application.



In case you want to get rid of the task instances simply enter ‘end’ and complete the task once again. If the condition at the exclusive choice gateway works correctly the process should come to its end.

But… How Does the Magic Work?

Without going into too much detail let’s spend some closing words how the complete lifecycle works:

As soon as someone opens the task execution UI the referenced Web Dynpro component will be initialized and embedded into it. In addition the Web Dynpro context will be enriched with the data that was defined in the input mapping of the task. That’s for example the reason why the first task shows “<Enter some value>” as this literal has been defined as start value for the data object in the process.

By defining the GACProperty in the Web Dynpro view FlashIsland will automatically propagate the bound value from the Web Dynpro context to the variable in the Flex application which has the same name (*myProperty*). As this variable was bound to the Flex text input you could immediately see it when the application was rendered (and registered itself to FlashIsland).

Remember that we declared the variable as bindable? This has the effect that a change is automatically propagated using a generated event. Flash Island listens to such events and keeps the connected Web Dynpro context attribute in sync.

Finally by defining the GACEvent we created an event channel back from Flex to Web Dynpro. As soon as Flex fires an event with the name that was specified in the properties of the GACEvent (*flexComplete*) the Web Dynpro event handler can react on it. In our case this triggered the *complete()* method at the Component Controller which subsequently fires the Web Dynpro c*omplete* event.

In the task editor of SAP NetWeaver BPM we specified that this event indicates the completion of the task. Thus whenever it is fired by Web Dynpro and caught by the framework it puts the task to completed, reads the data of the Web Dynpro context and writes it back to the process context via the output mapping.

Et voila…

Related Content

SDN Homepage

SAP NetWeaver BPM @ SDN

SAP NetWeaver CE Library

Adobe Flex Builder

Adobe Flash Player

Copyright