EDM EMM Scenario Part IV: Process Modeling and Portal Integration

Applies to:

Frontend:
• Microsoft Internet Explorer 6.0 SP1 or higher

Summary
This tutorial is a part IV of a series that describes core concept of composite applications and how to build them with NW04s based upon the EDM demo scenario, as been shown in several instances by the Market Development Engineering team.

The focus of this document lies on the process modeling with Guided Procedures and Enterprise Portal integration.

Author(s): Mark Mauerwerk, Hans-Joachim Odlozinski
Company: Axentiv AG
Created on: 16th January 2007

Author Bio
Mark Mauerwerk is currently working as Solution Architect of SAP Platform Ecosystem primarily focused on developing early stage demos and technical due diligence of SAP’s newest technologies. He is an expert in SAP NetWeaver architecture (ABAP and JAVA) and senior consultant at axentiv AG, a German SAP service partner and ISV enabling partner. He looks back on many years of experience in development and consultancy both internally at SAP as well as for customers & partners in the field.

Hans-Joachim Odlozinski is an expert in SAP application development architectures at axentiv AG, a German SAP service partner and ISV enabling partner. He looks back on many years of development and consulting both internally at SAP as well as for customers & partners in the field. He is currently working as Solution Architect at the Market Development Engineering team of SAP Platform Ecosystem. At axentiv AG he is team manager for Architecture & Development.
## Table of Contents

Introduction ...................................................................................................................................... 3
  Introduction to Composite Application Framework Guided Procedures (GP) ............................... 3
  Required Authorizations............................................................................................................... 4

Implementation .......................................................................................................................... 5
  Prerequisite: Creating the Portal Business Package .............................................................. 5
  Process Overview .................................................................................................................... 9

Create a Process ....................................................................................................................... 11
Create the Actions .................................................................................................................... 12

Attach Callable Objects to the Actions .................................................................................. 12
  Create callable objects of type iView ................................................................................. 12
  Attach Existing Callable Object .......................................................................................... 14
  Create Callable Object of Business Type “Business Logic” ................................................ 14
  Create callable object of business type “Composite Application Service” ......................... 15
  Assign Targets of the Result States .................................................................................... 16
  Assign Default Values to Input Parameters ......................................................................... 16

Consolidate Parameters ............................................................................................................. 17
Assign Roles .............................................................................................................................. 17

Testing ........................................................................................................................................... 19
  Test Callable Objects ........................................................................................................... 19

Related Content ............................................................................................................................. 20
Copyright........................................................................................................................................ 21
Introduction

This document is the final part four of the implementation HowTo's for the EDM scenario. Since you already have been enabled your backend services, implemented additional business logic in the Service Composition layer and modeled your UI endpoints, you now will integrate the content in your portal environment and finally model the process in the GP design time.

Prerequisite for this part is that you have already implemented all steps of part I-III of this document series.

Figure 1: Development procedure of EDM Scenario – Section IV: Process Modeling and Portal Integration.

As the UI modeling all steps take place in SAP Enterprise Portal.

Within the context of our EDM scenario you will learn how-to:

- implement a process in the GP design time
- integrate the content in your portal environment

Introduction to Composite Application Framework Guided Procedures (GP)

Guided Procedures is part of the SAP NetWeaver usage type EP and is a framework for modeling and managing processes that involve access to multiple backend systems.

For designing workflows and their building components, GP offers a Web-based design time toolset. There are numerous distinct applications and services available for use in GP processes by exposing them as callable objects. Callable objects are an abstraction for various service endpoints as for instance the application services and the iViews we have already implemented in the previous steps. This abstraction principle enables the composition of those existing diverse services and applications which are used to build actions and blocks of the process logic.

Once created in the design time, processes can be initiated and executed using the GP runtime. The access to tasks and resources assigned to individual users is role-based. GP uses a generic Java workflow engine and also integrates with the Universal Worklist (UWL) capabilities.
Additionally GP delivers administrative functions to execute a number of configuration tasks, such as setting up endpoint aliases, creating e-mail templates, and so on.

**Required Authorizations**

There are three major roles for guided procedures to execute guided procedures actions, design guided procedures, and performing administrative tasks such as monitor process instances, created end points to systems, and manage forms. You will also need the standard user role if you want to see guided procedure tasks in your Universal Worklist (if you want to use the UWL you need to configure the Guided procedure Connector, see https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/3605 for more details).

The business process expert will probably need the following roles:

- GP Administrator
- GP Business Expert
- GP User
- Standard User Role

The business process expert would then see the top level navigation of Guided procedures as depicted in the screenshot below.
Implementation

Prerequisite: Creating the Portal Business Package

As we want to deliver the Event Planner application as well as the ShowSummary iView (VisualComposer iViews EventAdmin and ShowSummary) as a SAP NetWeaver Portal business package in the role of an ISV who wants to later certify the complete composite application as a Packaged Composite application, we have to follow some guidelines from the SAP Integration and Certification Center (http://sdn.sap.com → Partners and ISVs → Integration and Certification). Here is the download address for the Portal Content Structuring Guidelines:


Please take some time to read this document as it will save you a lot of bother at certification time to have followed these instructions!

We assume that you have a basic knowledge of using the Portal Content Administration and proceed accordingly (for a detailed documentation see: http://help.sap.com/saphelp_nw2004s/helpdata/en/6b/a7a62102194ad0879b19731905d2de/frameset.htm):

Create a portal content folder Portal Content/Content Provided by Other Vendors/END User Content/edm.com:Event Materials Management:

Create subfolders with names iViews, Roles, Pages, Worksets. Then create a portal role EventPlanner in the newly created subfolder Roles:
After finishing the role creation wizard, follow the dialog to open the object editor. Ensure, that the role property Entry Point is set to yes. Then save but leave the object editor opened.

Create a second role GPProcessParticipant in the subfolder Roles. Open the object editor and set the role property Invisible in Navigation Areas to true (This is a "hidden" role which will not reflect in any content reachable via the portal top-level navigation. It will simply later "contain" the GP relevant iView ShowSummary). Then save and close the object editor.

Create a top-level workset EDM EMM in the previously created subfolder Worksets:
Open the object for editing. Create a simple workset named *Event Materials Management* in the same subfolder *Worksets* and open it for editing.

Create a Page named *EDM 1.0* in subfolder *Pages* using the *Default Page Template* and adding the page design *1 Column (Full Width)* and close the wizard.

Right-click the page in the portal content tree and select *Add Page to Workset → Delta Link* to add the page to the simple workset previously created.

Close the simple workset *Event Materials Management*. Right-click it now in the portal content tree and select *Add Workset to Workset → Delta Link*.

Close the complex workset *EDM EMM* (Only the object editor of role *EDM Event Planner* should now be open). Right-click it in the portal content tree and select *Add Workset to Role → Delta Link*. Close the object editor of role *EDM Event Planner*.

Now the general content structure has been set up. If you assign a user to the portal role *EDM Event Planner* the composite content should show as an empty structure of the following form:

Copy the Visual Composer iViews (location in the portal content: *Portal Content/Visual Composer/Models/enterpriseSOAToday/iviews*) to the previously created iViews subfolder of your portal business package.
Edit the iView’s permissions and add the `everyone` role to the Assigned Roles lists and allow end user access:

- Open the hidden role `GPProcessParticipant` and assign the iView `ShowSummary` to this role.
- Open the portal page EDM 1.0 and assign iView `EventAdmin` to this page.
- For test purposes create a portal user that you assign to role `EDM Event Planner`. Additionally for later use in GP create two users that represent the business process roles `EDM Manager` and `Purchase Manager` respectively. Assign the UME roles `GP Runtime WC`, `GP User`, and `Standard User Role`, and `GPProcessParticipant`.

Open the hidden role `GPProcessParticipant` and assign the iView `ShowSummary` to this role.
Open the portal page EDM 1.0 and assign iView `EventAdmin` to this page.
For test purposes create a portal user that you assign to role `EDM Event Planner`. Additionally for later use in GP create two users that represent the business process roles `EDM Manager` and `Purchase Manager` respectively. Assign the UME roles `GP Runtime WC`, `GP User`, and `Standard User Role`, and `GPProcessParticipant`. 
Process Overview

The business process of our scenario has been described in the overview document and has been depicted already on an informal level as follows:

Process modelling with GP generally comprises the following activities:

- Defining the components that build the process
- Defining the order for executing these components
- Defining and consolidating the roles for the contributors in the process
- Creating the process template itself

Depending of your design approach you may start bottom-up by defining the callable objects at first or top-down by defining the process template at first. Since we are starting from a particular process model we choose the top-down approach here.
The main process building elements are:

- The process template, which defines the process context, that is, the input, output, and configuration parameters.

- Blocks are the main building elements in a process template. You can embed other blocks and actions in a block. The items in the block can be executed sequentially or in parallel, and you can also define ad-hoc items and loops, which are executed in the event of certain conditions occurring. The top-level blocks in the process template form process phases. They are always executed in sequence.

- The individual steps in a process in GP are implemented as actions. They contain at least one callable object, which enables you to use external services and applications in processes. Actions also contain additional metadata to define the behaviour of the callable objects within the process itself.

- With Result States you can associate transition targets for the callable objects of the process. This provides the flexibility of process logic and allows you to guide process flow based on runtime decisions.

The process building elements of our scenario according to the scenarios process model are depicted in the picture above.
All process components are stored in a folder in the so-called gallery, which represents the repository for the design time elements.

The gallery opens when you enter the design time from the top level navigation.

The first step is to create a folder for the scenario:
1. Click “Create Folder”
2. Enter the name “ESA Today” and a description for your process folder

Create a Process
1. Select your process folder and choose Create Process to create a new process template. The process consist of two main parts: a table view to model the flow logic of the process and a tab view to edit the meta-data of the process (see screenshot below)

2. Enter name EventAdministration and a description of your process and save.
3. As Block is already selected in the item list, simply click icon to create a new sequential block.
4. Mark the new line with the block entry and edit the name and a description for the block.

**Create the Actions**

As with step 4 of the process definition above proceed to define the actions by simply clicking to create an action. Select the action and enter name and description according to the table below.

According to the process model we have the following actions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Callable Object</th>
<th>CO Type</th>
<th>Result state</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDMApproval</td>
<td>ShowApprovalCO</td>
<td>iView</td>
<td>Reject</td>
<td>PurchaseOrderRejected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Approve</td>
<td>PurchaseApproval</td>
</tr>
<tr>
<td>PurchaseApproval</td>
<td>ShowApprovalCO</td>
<td>iView</td>
<td>Reject</td>
<td>PurchaseOrderRejected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Approve</td>
<td>1. MapEventId</td>
</tr>
<tr>
<td>MapEventID</td>
<td>MapEventIDCO</td>
<td></td>
<td>business logic</td>
<td>default</td>
</tr>
<tr>
<td>CreatePurchaseOrder</td>
<td>CreatePurchaseOrderCO</td>
<td>App Service</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PurchaseRequisitionHandler, convertPRToPOForEvent)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PurchaseOrderApproved</td>
<td>ShowSummaryCO</td>
<td>iView</td>
<td>Complete</td>
<td>2. MapEventID</td>
</tr>
<tr>
<td>PurchaseOrderRejected</td>
<td>ShowSummaryCO</td>
<td>iView</td>
<td>Complete</td>
<td>Terminal</td>
</tr>
<tr>
<td>MapEventID</td>
<td>MapEventIDCO</td>
<td></td>
<td>business logic</td>
<td>default</td>
</tr>
<tr>
<td>RemovePurchaseRequisition</td>
<td>RemovePurchaseRequisitionCO (PurchaseRequisitionHandler, removePurchaseRequisitionForEvent)</td>
<td></td>
<td>App Service</td>
<td>none</td>
</tr>
</tbody>
</table>

Table 1: Process Definition.

Note: To create an action the related block has to be switched to the edit mode ( ). The order of the actions in the process definition represents the order of the actions at runtime also. This order can only be influenced dynamically by the definition of result states and their related targets at the definition of callable objects. To range in an additional action in the process hierarchy, just create it, select it and move it to the right position in the process table by using the move-up arrow icon.

**Attach Callable Objects to the Actions**

If already implemented at design time of the process template, you can define the callable objects on the fly after definition of an action by simply click from the process template or later directly in the gallery by clicking “Create Callable Object”. Both will start a wizard for the definition of the callable object.

Referring to the development process we have a synchronisation point of the overall development process here. Up to here the services and iView may be developed in parallel. But from now on all interfaces and related UI’s should be available with interfaces of status released.

According to the process definition above we have to define 4 callable objects of 3 different types (see table above).

**Create callable objects of type iView**

1. If you started from process template, once you have created and saved the corresponding action, click to start the wizard “Create Callable Object”.

SAP DEVELOPER NETWORK | sdn.sap.com
BUSINESS PROCESS EXPERT COMMUNITY | bpx.sap.com

© 2006 SAP AG
2. Basic data: Type a name (ShowApproval) and description and select the callable object type “User Interface portal iView or page” and click Next.

3. Define Callable Object: Select the portal role and the desired page or iView of that role. In our case it is the role “GPProcessParticipant” and select “ShowSummary”.

4. Define Input: leave the default parameters as designed in the visual composer and click Next.

5. Define Output: proceed according to step before.

6. Set Configuration: Here you define the result state buttons of your action. Add the buttons as depicted below and click Next.

7. Finish: Click Finish to confirm your edits.
Attach Existing Callable Object

Once created a callable object may be reused by another action. In our scenario it would be directly the next one “PurchaseApproval”.

1. Click the insert icon to open up the gallery in an additional page below.
2. Once the filter is already set for callable objects it’s easy to navigate to the callable object you have created and click Select.

Create Callable Object of Business Type “Business Logic”

You use the Business Logic callable object type to implement dynamic expressions depending on the actual values of the input parameters. Such expressions can be defined for output parameters without any coding or use of an external rule engine. In our scenario we need a type mapping from Double to Long between two service calls.

You will use the same wizard as already described for callable object of type iView above, but there are different configurations (Action MapEventID):

1. Basic data: Choose type “Process Control, Business Logic”.
2. Define input: Type the values according to the picture below.

3. Define Output: see below.
4. Set Configuration: On the Set Configuration screen, enter the expression for the output parameter defined previously by selecting the “...” button. In the formula editor type `ROUND(@doubleIn)` and validate for typos. At least you need one result state. Add one with name=default and expression=true (configuration see below) press Next and finish.

Create callable object

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous</td>
<td>Next</td>
<td>Cancel</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Configuration**

<table>
<thead>
<tr>
<th>Output Parameters</th>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>longt</td>
<td><code>ROUND(@doubleIn)</code></td>
</tr>
</tbody>
</table>

**Result States**

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>true</td>
</tr>
</tbody>
</table>


The same mapping action have to be defined a second time to map the event ID again (see Table 1: Process Definition.).

Create callable object of business type "Composite Application Service"

The action CreatePurchaseOrder calls one of our Composite services.

1. Start the wizard according the steps above.
2. Basic data: Select Type “Service, Composite Application Service” and name CreatePurchaseOrderCO.
3. Define Object: Navigate to your Composite Application, select the Service and the method according to screenshot below.
   - Leave the rest as default and Finish.

According to the process definition (Table 1: Process Definition.) we need a second callable object of this type for the final action. Use the values of Table 1 for the definition of the callable object.
Assign Targets of the Result States

After all actions have been designed and all callable objects have been defined you have to assign the targets for the result states of the callable objects (the block must be in the edit mode):

1. In the process template select the line of the result state
2. For the first result state Reject in action EDMA_approval select “PurchaseOrderRejected” as depicted in the screenshot below.

**Block: EventAdministration**

![EventAdministration Block Diagram]

1. In the process template select the line of the result state
2. For the first result state Reject in action EDMA_approval select “PurchaseOrderRejected” as depicted in the screenshot below.

Proceed according to the process definition in Table 1.
Don't forget to save all your changes.

Assign Default Values to Input Parameters

What remains to be done is to set the input parameter approvalStatus of Callable Object ShowSummary so that we receive the correct message text during the process that either demands approval (approvalStatus="") or states approval (approvalStatus="true") or rejection (approvalStatus="false").

The normal procedure would now be to set the default value for this input parameter at the action level. One would select the action and would switch to its parameter tab. Here one would select the line Input Parameter (do not expand), press the button Default Value, and insert a single default value line with the correct value for the action at hand (approvalStatus=true for action PurchaseOrderApproved and approvalStatus=false for action PurchaseOrderRejected).

**Note:** Due to a current bug in SP10 this procedure does not work so we will show you a workaround. Note, that SAP will provide a patch for SP10 that includes a fix for this bug so that in this case the procedure described above should be applied.

**Workaround:**
Do not set the default values on the action level but on the block level instead. Again switch to the parameter tab, this time the one of block EventAdministration. In the parameter table first select the line for Input Parameters of action PurchaseOrderApproved. Now press the button Default value and insert a single default value line with approvalStatus=true and press the Set button. Repeat this procedure for the Input Parameters of action PurchaseOrderRejected with approvalStatus=false. This finishes the default value setting.
Consolidate Parameters

You must group the parameters, so that the output of the process actions is automatically entered as an input for the other actions. Consolidation of parameters can be performed at action, block and process level.

Procedure:

1. Select block "EventAdministration" and choose tab Parameters.
2. Switch to edit mode.
3. Mark all parameters eventId and doubleln of type double holding down the CTRL key (this include those as part of the input- and output structures also) and click on button Map. In the Name field, enter the name "eventId" for the group and choose Create.
4. Proceed as in the step above for all eventId and longIntOut parameters of type Long Integer and consolidate the to group "eventIdLong".
5. According to step 4 and 5 group the poNumber parameters of type String into the group "poNumber" (Note: the Response parameter of action CreatePurchaseOrder belongs to this group, also).
6. Uncheck all checkboxes is Input except for group eventId and save all process data.

Finally you should see the following List of context parameters in the block:

Refer to http://help.sap.com/saphelp_nw2004s/helpdata/en/de/598041a17e060de10000000a1550b0/frameset.htm for more details.

Assign Roles

For each action of a block, GP creates a separate process role. These roles are further propagated from block to process level. At runtime, you assign user(s) to each existing process role to define who is responsible for its execution.

Consolidating roles means grouping and displaying them as a single one. You consolidate the roles for those actions that must be executed by the same user(s).

1. In the gallery, select the block “EventAdministration", and open the object’s design time.
2. Switch to edit mode and select tab Roles.
3. All available roles are displayed. You can check for which action or block they are defined in the column Items.
4. To consolidate the roles: Select the roles from the list, enter a name for the consolidated parameter in the field Consolidate To and press button Go.
5. Repeat the step before according to the configuration below:
Save and activate all data changes. On process level you need to define the role types according to the configuration below.

Assign portal users to the default roles:

1. Choose tab Default Roles.
2. Press Add Default and select your previously created portal user corresponding to the role (Processor of EDMApproval corresponds to the business process role EDM Manager).

Repeat this step for all the default roles.

Finally save and activate all your changes.
Testing

Test Callable Objects

The Guided Procedures framework enables you to test your standalone callable objects.

1. In the gallery, select the callable object that you want to test, and choose Open… to open its design time.

2. Choose tab Test.

3. If the callable object defines required input parameters, enter values for them.

4. Choose whether you want to test the object in execution or in display mode. Note: if the object is tested in display mode, you only get a preview of the function, but you cannot execute it.

5. Choose Execute. If the application has a user interface, it is displayed in the Callable Object Execution step. To display the test results for callable objects with user interface, choose Complete Step. This step is skipped if the callable object is executed in the background, and test results are displayed directly.
Related Content

- Composite Application Community: https://www.sdn.sap.com/irj/sdn/developerareas/caf
- Online Help: http://help.sap.com/saphelp_nw2004s/helpdata/en/33/198141f906040de10000000a1550b0/frameset.htm
- BPX SDN Community for business process experts: http://bpx.sap.com/