Advanced Techniques with Guided Procedures in SAP NetWeaver Composition Environment 7.1

Applies to
SAP NetWeaver Composition Environment 7.1, SR3 – also the SDN downloadable SAP NetWeaver CE 7.1 trial version can be used for working this exercise.

Summary
This tutorial explains advanced capabilities of the process modelling tool Guided Procedures that is part of SAP NetWeaver CE 7.1. The features are explained with the help of business case for the creation of a business partner. Learn here how a decision is modelled within Guided Procedures and how this influences the process flow. Additionally the document depicts the use of external services in a Guided Procedures business process.

Authors:  Belinda Lineman, SAP America
Ginger Gatling, SAP Labs LLC
Matthias Kretschmer, SAP AG;

Created on: August 2007

Author Bios

Belinda Lineman
is a SAP Senior NetWeaver Education Consultant. Her area of concentration is SAP NetWeaver CE and Business Process Integration. She is a subject matter expert for SAP NetWeaver Process Integration. Belinda is responsible for educating customers and for knowledge transfer, enabling consultants to obtain certifications for SAP NetWeaver Process Integration both technical and developer, SAP NetWeaver WebAS Technical Administration, and SAP NetWeaver WebAS Security. She has been with SAP since 1996. Belinda has focused on integration scenarios, system administration, and IT infrastructures.

Ginger Gatling is a SAP NetWeaver product manager for business task management (universal worklist, collaboration tasks, guided procedures, and SAP Business Workflow), SAP NetWeaver Exchange Infrastructure, SAP NetWeaver Mobile, and SAP NetWeaver Master Data Management technology. In this role, she is directly responsible for all product-related questions as they pertain to rollout and roll-in to and from the customer as well as the SAP Americas field organizations. Ginger has been with SAP since 1997. She has focused on system administration and integration topics such as workflow, security, ALE, IDOCs, CRM Middleware, and administration of SAP Web Application Server.

Matthias Kretschmer joined SAP in September 2001 as a BA Student (Information Technology). During his studies Matthias had the opportunity to gain an insight into several departments within SAP, e.g. Mobile Infrastructure, Technology Consulting and MAX DB. Matthias successfully completed his studies in September 2004. From 01.10.2004 until 31.07.2006 Matthias worked in the NW ESI F XPI department (XI Runtime, Monitoring) as a software developer (Java, ABAP). At moment Matthias works as a product specialist for the NetWeaver Demo Team.
Table of Contents

Introduction to exercise.......................................................................................................................................3

Prerequisites .......................................................................................................................................................5

Step 1: Build Process and Main Block ..............................................................................................................6

Step 2: Build Project Block and First Test .......................................................................................................21

Step 3: Build Approval Block ..........................................................................................................................50

Step 4: Parameter Mapping and Role Assignment for Approval Block ............................................................76

Step 5: Parameter Mapping for the Process .....................................................................................................87

Step 6: Role Assignment and Final Setup for the Process ..............................................................................91

Step 7: Test .......................................................................................................................................................95

Optional step 8: Use a Service from the SDN Service repository .................................................................101
  1. Locate the web service on sdn. ..............................................................................................................101
  2. Test the web service in the web services navigator. ..............................................................................101
  3. Look at the logical destination created for this service in the NetWeaver Administrator .....................101
  4. Create a callable object that uses this web service. ...............................................................................102

Optional step 9: Ad Hoc Steps .......................................................................................................................102
  1. Create an action and callable object called AC_RevenueReport and CORevenueReport ...................102
  2. Update the AC_InputBPInformation step in your process to have an ad hoc action. ..............................103
  3. Add the ad hoc view to the runtime view and test. ...............................................................................103

Copyright.........................................................................................................................................................104
Introduction to exercise

In this exercise you will build a process related to the creation of business processes. Business partner information is submitted, if the partner is for the region EMEA, a project must be created. All business partners must follow an approval process. This process will be implemented in 3 blocks. There will be parameter mapping in each block as well as parameter mapping across all the blocks. The outline of the process is as follows:

- **Main block** – this block is always executed, it starts the process by collecting basic business partner information. Check out screencam [Build a GP Process with two steps](#) (7 min).
  - Input business partner information, including the region (this is a search by field)
  - Check the region (is it EMEA?) Skip to the appropriate block, based on the region.

- **Project block** – This block is only executed if the region is EMEA. See recording with voice [Enhance the GP process with a second block](#) (7 min).
  - Enter project – Here you enter basic information about a project and select a user to approve the project. The user returned should be used for the approval project step.
  - Approve project – This step routes to the person selected from the previous step and they approve the project.
  - Create project – This step creates a project on the local database.
  - Deploy and test the process, refer recording: Assign roles and test the process (6 min).

- **Approve Block** – This block is always executed. If the region is not EMEA, it is executed after the Main Block. If the region is EMEA, the project block happens, then the approve block. A final display will include information from the project block. Refers to movie Add approve block (14 min).
  - Add additional information – In this step you can change or enter additional business partner information.
  - Pick user – In these steps we will select a user for the approval. In the Project Block this is done via a Visual Composer application, in this block we will do it using specific guided procedures callable objects to select a user at runtime. In this step you pick the user who will do the approval.
  - Assign user to role – This step assigns the user selected in the previous step to the subsequent step (the Approval step).
  - Approval – This step does the actual approval of the business partner.
  - Create the business partner in ERP – Creates the business partner in an ERP system by calling a web service that calls a BAPI.
  - Get the business partner details – In this step you call another web service that gets the details of the business partner you just created (included the business partner number).
  - Display business partner – You will create you own Visual Composer application to display the business partner details. This will also include project details if the project block was executed.

Parameter mapping is extremely important in this exercise. Please take your time and ensure you understand why you are mapping the parameters in a particular way. Check screencam Add VC UI and complete process (9 min).
The following picture makes clear the business process visually:

**Initiate the creation of new business partners, process varies depending upon the region**

Business partner created in application system

- Project Steps for EMEA partners
- Approval Steps for all partners
**Prerequisites**

You need a running SAP NetWeaver Composition Environment 7.1 SR3 system. Additionally you need the design tools like a "normal" browser (IE or Mozilla). Installed and configured SAP NWDS would be a benefit but not needed.

This exercise re-uses some prepared components that are built within the exercise "CE Project Example" that can be downloaded from SDN. So it’s to have done this “CE Project Example” exercise beforehand or at least to do the parts of the exercise that produce the re-used components. The re-used parts are:

a) Visual Composer UI for the “Program Lead” that is built in chapter “Building the Visual Composer User Interfaces” (starting page 13, steps 1 - 45)

b) We need a Visual Composer UI for the “Program Lead” that is built in chapter “Building the Visual Composer User Interfaces” (starting page 30, steps 46 - 75).

c) Local web service that offers the possibility to create a project data set onto the local database. The creation of this web service is explained in chapter “Building the Business Objects”, starting page 41, steps 1 – 44.

Additionally this tutorial uses Actions and Callable Objects that were built in another exercise “CE GP Example”. These are the components:

- **AC_ReviewBPInformation** (review the existing business partner information with the option to change or add a secondary search term)
- **AC_PickUser** (Select a user that will be used to approve the business partner)
- **AC_AssignUserToRole** (Assigns the user from the pick user step to the next step)
- **AC_ApproveBusinessPartner** (Approve the business partner)
- **AC_CreateBPinERP** (Creates the business partner in the ERP system)
- **AC_GetBPDetails** (Reads the business partner details from the ERP system)
- **AC_DisplayBP** (Displays the business partner we just read from the ERP system. This will be done by a Visual Composer application you will create.

So it’s good to have worked the tutorial “CE GP Example” beforehand.

You need at least two users for executing this exercise: one for requesting the creation of a new business partner and second for the approval of this request. These users have to be assigned to the corresponding roles during the design time. We use here the users “CE251” for the requestor and “fallf” for the approver.
Step 1: Build Process and Main Block

The process has three blocks. In this task you will build the process and the main block. See also screencam Build a GP Process with two steps (7 min).


2. Click on Design Time.

3. Select Create Folder to create a new folder for all your objects.

4. Enter an appropriate name and description.
5. Click on Create Process

6. Choose English and click on Create
7. Provide appropriate process name and description.

8. Create a new block by clicking the new icon.

9. Choose sequential block and click create.
10. Highlight the block that was created and provide it with the following name and description

Name: **MainBlock**

Description: **Main block**

Then click **Save All**.

11. Create 2 New Actions by clicking (2 times) on the **Create New** icon after highlighting the block

12. The result should be as shown in the diagram.

**NOTE:** It is OK if your default action names do not exactly match the diagram.
13. Click on the first Action and give it the following name and description:

Name: **AC_InputBPInformation**
Description: *input information for the business partner*

*Hint*: After entering a name and description you can see the name in the process line by clicking on the action line.

14. Click on the second Action and give it the following name and description:

Name: **AC_CheckIfEMEA**
Description: *check if the business partner is in region EMEA.*

Select **Save All**.

15. Add the callable object for **AC_InputBPInformation**. Select the action, **AC_InputBPInformation**, and select “create” to create a callable object.
16. Enter the following information:

   **Name:**
   CO_InputBPInformation

   **Description:** Input the information for the business partner

   Select *Form → Web Dynpro Form* for the Type of callable object.

   Then select *Next*.

17. Select *Add → Input Field* to enter a field for our Web Dynpro form.
18. Enter **FirstName** for the Input Field ID and the Input Field Label and select **Create**.

19. Repeat the previous two steps for the fields **LastName** and **SearchBy**.
20. Select Next. Then select Finish.

21. Now create the callable object which will check if the value “EMEA” was entered in the SearchBy field. Select the action AC_CheckIfEMEA and select Create to create a callable object.
22. Enter the following information:

**Name:** CO_CheckIfEMEA

**Description:** check if the region is EMEA

Select Process Control → Business Logic for the callable object Type.

Select Next.

23. Select Next until you get to Define Input. Select Insert New.

24. Replace the default name, *Parameter1*, with the name the SearchBy for both the Name and Technical Name fields.

Select Next until you get to Set Configuration.
25. In the *Set configuration* area do the following:
   
a) *Add* under **Result States**.
   
b) Provide the name *isEMEA*
   
c) *Select AddState*.

26. Select the “…”
27. Enter the expression:

```javascript
@SearchBy=="EMEA"
```

Then select Validate and Apply.

28. You should now have the expression like the screenshot.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEMEA</td>
<td>@SearchBy==&quot;EMEA&quot;</td>
</tr>
</tbody>
</table>
```

29. Repeat the existing steps creating a new state: isNotEMEA.

The expression should read:

```javascript
@SearchBy!="EMEA"
```

Select Next and Finish.
30. Select the Test tab from the callable object.

31. Test the callable object by entering **APA** in the SearchBy field and select **Execute**.
32. The returned value should be `isNotEMEA`.

33. Select **Start Over** to repeat the test entering **EMEA** for the **SearchBy** and the return value should be **isEMEA**.

Select **Save All**.
34. Now we need to do parameter mapping for the **MainBlock**. Select your **MainBlock** and select the tab **Parameters**.

Notice the parameters we have:

- **FirstName**
- **LastName**
- **SearchBy** (from the input step)
- **SearchBy** (from the check if EMEA step)

We need to map both of the **SearchBy** fields so that the value provided in the input step will be evaluated to see if it is EMEA.

35. Highlight the first **SearchBy** field.

36. Press the <CTRL> key and highlight the second **SearchBy** field. Now both fields should be highlighted.
37. Select Group and provide the name GRP_Main_SearchBy. We will do a lot of parameter mapping in this exercise, so we will use the convention GRP_<Block>_<value>.

38. You will now see your group with the two parameters included in the group.

Select Save all.
Step 2: Build Project Block and First Test

In this step you will build the project block and do a first test of your process. Refer recording with voice 
Enhance the GP process with a second block (7 min).

1. Select your process and select create Block.

2. Select Sequential Block and name the block and description ProjectBlock. Your process should look like the diagram.
   Select Save all.

3. Create three new actions by clicking (three times) on the Create New icon after highlighting the block. Your process should then look like the diagram.
   Note: Your actions may have different numbers, such as Action4, Action5, and Action6. That is fine. For now just ensure you have three actions below your project block.

4. Rename each of your actions so that they have the following names in the following order:
   
   Name: AC_EnterProject
   Description: enter the project information

   Name: AC_ApproveProject
   Description: approve the project

   Name: AC_CreateProject
   Description: create the project on the local database
5. Select the action **AC_Enter Project** and select to create a callable object.

6. Enter the following information:

   **Name:** CO_EnterProject
   **Description:** Enter the project
   **Type:** User Interface → WD4VC Application

   This step uses a Visual Composer application.

   Select Next.
7. In the following screen you will get a list of Visual Composer applications. The application we need is created within another exercise: "CE Project Example". We need a Visual Composer UI for the "Program Lead" that is built in chapter "Building the Visual Composer User Interfaces" (starting page 13, steps 1 - 45).

Select Next.

8. Notice the input parameters for this Visual Composer application.

Select Next.
9. Notice the Output_Parameters. When you open up the output parameters, notice that a login is returned. We will use this value at runtime to route the approval step – which is the next step in the process.

Select Next, Finish, and Save All for your process.
10. Test the callable object by selecting your object and select the Test tab. Then select **Execute**.
11. You will see a test screen where you can enter test information like in the diagram. Select fallf for the user and then select Submit in the Visual Composer application.
12. In the return values you can see that the login *fallf* was returned from the Visual Composer application.
13. Repeat the previous steps to create a callable object for the action `AC_ApproveProject`. Please use the following information:

**Name:** CO_Approve_Project

**Description:** approve the project

**Type:** User interface → WD4VC

Please choose this exact Visual Composer application. The application we need is created within another exercise: “CE Project Example”. We need a Visual Composer UI for the “Program Lead” that is built in chapter “Building the Visual Composer User Interfaces” (starting page 30, steps 46 - 75).

Select Next.

Note: Optionally, notice the input parameters which are Title, Description, Comment, Schedule, First Name, Last Name, and Login.

Select Next until you can select Finish.
14. OPTIONAL: Test the callable object by selecting the callable object then select the Test tab.

Select Add Row.
15. OPTIONAL: Enter some test data like what is in the diagram and select Execute.
16. OPTIONAL: You will see the Visual Composer application with your test data. Select Approve.

17. OPTIONAL: You can see the result from the callable object execution is approve.

Select Save all.
18. Return to your guided procedure, select the action **AC_CreateProject** and select to create a callable object.

19. Enter the following information for your callable object:

   **Name:** CO_CreateProject

   **Description:** create the project using a service on the local CAF layer that updates the database

   **Type:** Service → Web Service

   Select Next.

20. Use the URL for the web service that creates a project instance in the local database. This web service is built within the exercise “CE Project Example”, chapter “Building the Business Objects”, starting page 41, steps 1 – 44. In this case it’s the URL shown in the screenshot, however you’ve to use your URL (= your web service).
21. Select Go.

You will then see the properties of the web service.

22. Select Create and you can see the input parameters for the service. Optionally, tab to look at the Return parameters.

Select Next.

As you select Next you can optionally look at the input and output parameters. Select Next until you can select Finish.
23. Test the callable object by selecting the callable object and selecting the *Test* tab.
24. Enter some test data and select **Execute**.
25. Notice that the call is successfully completed and you have a *key* returned to you.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>26b07a2c7c0-467f-11dc-98c0-001558c33404</td>
</tr>
<tr>
<td>createdBy</td>
<td>Guest</td>
</tr>
<tr>
<td>createdAt</td>
<td>2007-08-09T15:51:20.000</td>
</tr>
<tr>
<td>modifiedAt</td>
<td>2007-08-09T15:51:26.000</td>
</tr>
<tr>
<td>modifiedBy</td>
<td>Guest</td>
</tr>
<tr>
<td>title</td>
<td>test project</td>
</tr>
<tr>
<td>description</td>
<td>new bp</td>
</tr>
<tr>
<td>schedule</td>
<td>2 weeks</td>
</tr>
<tr>
<td>firstname</td>
<td>Franco</td>
</tr>
<tr>
<td>lastname</td>
<td>Fall</td>
</tr>
<tr>
<td>comment</td>
<td>test project</td>
</tr>
</tbody>
</table>

Result

- **Result**: Completed successfully

Output Parameters

26. Your *ProjectBlock* should now look like the one in the diagram. You have three actions and three callable objects.

We will now begin parameter mapping for the *ProjectBlock*.

Select **Save all**.
27. Select the **ProjectBlock** and select the **Parameters** tab.

28. Before doing the parameter mapping let’s look at what we need. The **Output** from the **AC_EnterProject** should be passed to the **Input** of both the **AC_ApproveProject** and the **AC_CreateProject**.

Here is an example of how the **Title** mapping will look:

Notice we are using the **Output** of the **AC_EnterProject** and mapping it to the **Input** of **AC_ApproveProject** and **AC_CreateProject**.
29. Here is an example of the mapping of the Description and a mapping of the Login. Notice that the Login is from the Output of the AC_EnterProject to the Input of the AC_ApproveProject. It is not used in the AC_CreateProject.

30. To begin the parameter mapping select the ProjectBlock and select the Parameters tab. Notice there are input and output parameters for each of the actions.

31. First we will map the Title field. To do this drill into the Output_Parameters of AC_EnterProject and select the Title field.

32. Scroll down to the Input_Parameters of AC_ApproveProject. Hold down the <CTRL> key and select Title.

33. Scroll down to the Input_Parameters of AC_CreateProject. Hold down the <CTRL> key and select
34. Name the group GRP_Prj_Title.

35. You will now see the group GRP_Prj_Title. It should include the fields listed in the diagram. The order is not important. It is just important all three fields are in the group.

36. Repeat the parameter mapping steps for the field Description. Name the group GRP_Prj_Description. The group should look like the diagram.

37. Repeat the parameter mapping steps for the field Comment. Name the group GRP_Prj_Comment. The group should look like the diagram.

38. Repeat the parameter mapping steps for the field Schedule. Name the group GRP_Prj_Schedule. The group should look like the diagram.
39. Repeat the parameter mapping steps for the field LastName. Name the group GRP_Prj_PjLastApprover. The group should look like the diagram.

40. Repeat the parameter mapping steps for the field FirstName. Name the group GRP_Prj_PjFirstApprover. The group should look like the diagram.

41. The login is a field that will be returned from AC_EnterProject and used to know how to route AC_ApproveProject. For this parameter mapping notice that login is not passed to the AC_CreateProject. It is only mapped between the enter project and approve project steps. The group should be named: GRP_Prj_Login.

42. Notice that one of the columns in parameter mapping is called Exposed In. This enables you to expose any fields where the value will be passed to the block.
43. You will notice many of the fields are marked as Exposed In. However, since most of our information is inside of the block, we don’t need them to be exposed. Deselect the Exposed In check mark for every row except GRP_Pri_Title and GRP_Pri_Description.

Select Save All.

44. Now we need to update our role assignment so that the AC_ApproveProject step is routed to the correct person. Select the ProjectBlock and select the Roles tab.

45. Select Processor of AC_ApproveProject and select the dropdown for the Filled from Context Parameter tab.
46. Choose \textit{Input\_Parameters $\rightarrow$ Login} as the parameter. This means says that the person to do the \texttt{AC\_ApproveProject} will come from the value of \textit{Input\_Parameters $\rightarrow$ Login}. Remember that the login is passed from the output of \texttt{AC\_EnterProject} to the input of \texttt{AC\_ApproveProject}.

47. We now need to say who will do each of the other steps. Select your process and select the \textit{Roles} tab. Notice we have a role for each step in our process, plus roles for \texttt{Administrator}, \texttt{Overseer}, and \texttt{Owner}. For our test, we will have the same person do each step listed. Since the actions will be done by the same person, we will consolidate the roles together.

48. Select each role that relates to your actions by using the \\texttt{<CTRL>} key to select multiple roles. Your roles highlighted should match the diagram.
49. In the field **Consolidate To** enter the value **BusinessUser**. Then select **Go**.

50. Select the drop down for the **Business User** and select **Initiator**.

51. Set all other roles to **Initiator** as well. Select **Save all**.

52. Select the process and select the **Parameters** tab. You will notice that many of the fields are marked as **Exposed In**. We do not need to pass any fields in before the process starts, so deselect each of these entries.
53. The Parameters tab should now have no fields marked as exposed.

Now select Save all and Activate.

You can now continue with assignment of roles and check the process, refer recording: Assign roles and test the process (6 min).

54. We are now ready to test your process. Select the process and select the Instantiation tab. Select Start Process Automatically. If we had parameters passed in before the process started, we would select Include default parameters. It is fine if you select it. Then select Generate Instantiate URL. Then select Open Instantiate Application to start your process.

55. Your process will launch in a new browser window. You can see the MainBlock and the Project Block. You can also see your first action to enter business partner information. Enter information (of your choice) and select Complete.
56. Notice we are now in the ProjectBlock executing the step to enter the project. Enter some information and select fallf from the Employee List.
57. Remember that the AC_ApproveProject is routed to the return value from the AC_EnterProject. You now get the message “Action is ready for processing by another user or role”. You can also see that Franco Fall is the processor for the AC_ApproveProject.

58. Launch a new browser and login to the portal (http://localhost:50100/irj) with the credentials for the user who is able to execute the next step. Here in this example the following credentials are true:

User id: fallf
Password: welcome123

59. Navigate to Guided Procedures → Runtime. Then select Tasks that require my action. There should be at least 1 entry.
60. Select your task, **AC_ApproveProject**. It is part of your process.

61. Notice that you have the project information passed from the previous step with the option to approve or reject.

   Select **Approve**.

   For now this ends our process. Let us go on and make a small change to our process. If the user selects **Approve** we will continue. If they select **Reject** we will have them enter the project again.

62. Return to your process in **Guided Procedures → Design Time** and select **Change all** to change your process. This will give you a new version for your process. When asked “Current version is active. Do you want to update to inactive version” say **Yes**.

   Drill into your **AC_ApproveProject** and look at the result states, **reject** and
63. Update the *reject* result state to point to **AC_EnterProject**.

64. **Activate** your process and start your test again using the *Instantiation* tab. Be sure to always select the appropriate user (here *fallf*) as the approver.

Once you execute the item as *fallf* select *Reject*. You will receive a message that the **Action has been completed**.
65. Return to the window of the user who started the process and select **Refresh**.

66. You will now see we have looped-back to the **AC_EnterProject** step. Enter some test data for the project fields, select the appropriate user (here: fallf) for the user. Return as user fallf, approve the project.

Congratulations! Your first test is now completed! Now…let’s move on to the Approval block!

Before continuing, close all test windows.
Step 3: Build Approval Block

In this step we will build the approval block. In this block we will reuse some existing actions and create other actions. Refers to movie Add approve block (14 min).

1. Return to Guided Procedures → DesignTime. Select your process, you can either select Change all or highlight the process and select Change for the process. You will get a new version for your process.

2. Highlight your process, create a new Sequential Block and provide the following name:

   Name: ApproveBlock

   Description: approve the business partner and update the ERP system
3. This step is purely informational, you will do the work in the next step.

Your ApproveBlock will include seven actions. Each action is listed below. However, you will not build all the actions. For some of the actions you will reuse existing actions that have been created for you in the CE251 session folder.

RESUSE EXISTING:
AC_ReviewBPInformation (review the existing business partner information with the option to change or add a secondary search term)

AC_PickUser (Select a user that will be used to approve the business partner)

AC_AssignUserToRole (Assigns the user from the pick user step to the next step)

RESUSE EXISTING:
AC_ApproveBusinessPartner (Approve the business partner)

RESUSE EXISTING:
AC_CreateBPinERP (Creates the business partner in the ERP system)

RESUSE EXISTING:
AC_GetBPDetails (Reads the business partner details from the ERP system)

AC_DisplayBP (Displays the business partner we just read from the ERP system. This will be done by a Visual Composer application you will create.)

4. Insert the actions by selecting your ApproveBlock and select the icon to Insert an action.
5. In the lower part of your screen, select the appropriate folder.

6. You will see many actions.

7. Select **AC_ReviewBPInformation**.
8. Select additional actions by clicking in the box next to the action. Select the following actions:

   - AC_ApproveBusinessPartner
   - AC_CreateBPinERP
   - AC_GetBPDetails

   AC_ReviewBPInformation is already selected, so you have four actions selected.

9. Press Select.

10. You will see the actions in the block. Now we need to get them in the correct order.

   There are up and down arrows to move the actions.

11. The order of the actions should be as follows:

    - AC_ReviewBPInformation
    - AC_ApproveBusinessPartner
AC_CreateBPinERP
AC_GetBPDetails

Optionally, after adding the actions you can test the callable objects that are part of the action.

12. We now need to add the other actions:
   AC_PickUser
   AC_AssignUserToRole
   AC_DisplayBP

   In the end, your actions will be in the order in the diagram.

13. Highlight your ApproveBlock and select the icon to create a new action.

14. Insert three new actions. The default action name does not matter. Now you should just have three additional actions.
15. Rename the three actions so they have the names as in the diagram:

- AC_PickUser
- AC_AssignUserToRole
- AC_DisplayBP

16. Now we need to get them in the correct order. To do this select AC_PickUser and use the up arrow to get the action to the correct place.

The AC_PickUser should be immediately after the AC_ReviewBPInformation.

17. Move the steps so they are in the order in the diagram.

Select Save all.

18. Select your action AC_PickUser and create a new callable object providing the following information:

- **Name:** CO_PickUser
- **Description:** Pick a user who will do the approval of the business partner
- **Type → User Management → Choose a User**

Select Next
19. Notice the output includes a *Unique ID* and *User Data*.

Select *Next* and select *Finish*.

20. Highlight the callable object, select the *Test* tab, and select *Execute*.

In this test, enter your user ID, in the *Find* field. Then select *Go*. 
21. You will see your user ID, select Choose.
22. The result will be a successful completion (here with user CE251) as the returned Unique Name.

23. Select your action AC_AssignUserToRole and create a new callable object providing the following information:

Name: CO_AssignUserToRole

Description: Assign the user selected in the previous step to the role for the next step. This is done in parameter mapping and role consolidation.

Type → User Management → Assign Users to Process Role

Select Next until you select Finish.
24. Select *Save all.*

25. We have one more callable object to create. This is for the final display of the business partner. In this display we want to include both the business partner information and project information. You will create the user interface for this callable object using Visual Composer. Launch the Visual Composer via [http://localhost:50100/VC](http://localhost:50100/VC)

   You need to login using your corresponding user ID (here: CE251) and password.

   The recording screencam [Add VC UI and complete process](http://localhost:50100/VC) (9 min) supports your work.

26. Once Visual Composer opens, select *Create new model* from the *Get Started* panel or from the menu bar under *Model.*
27. In the Development component line select New.

28. Enter an appropriate name for the new development component and select OK.
29. Ensure Composite View is selected for the Model Type.

In the Name field enter: **DisplayBPDetailsGrpxx** then select OK.

30. Select Start Point from Compose Model.
31. Drag and drop the **Start Point** to the canvas. The result should look like the diagram.

32. Right-click on the **start** and select **Define Data**. We are about to define the fields that will be passed from the guided procedures context to the Visual Composer application.
33. Select the + icon to add new fields as input parameters.

34. Select String.
35. Change the default field name to **BusinessPartnerNumber**.

36. Repeat the steps to add the following fields:

   - **FirstName**
   - **LastName**
   - **SearchBy**
   - **SecondarySearchBy**
   - **ProjectTitle**
   - **ProjectDescription**

   Select **Close**.

37. Click on the **start** and drag a line and release to see the context menu with **Form View**. Select **Form View**.
38. You canvas should look like the diagram.

39. Right-click on your form and select *Rename*.

40. Rename the form *Display Business Partner*.

41. Select the *Layout* tab at the bottom of the canvas.
42. You will now see the fields on your form.

43. Select the Businesspartner field and make the following changes in the Configure Input Field:

- **Label**: Business Partner Number
- **Read-only**: true
44. The result should change the layout to look like the graphic.

45. Repeat the steps for each field on the form. Updating the field to display only and providing an updated label. Your form should look like the diagram. You can make the field label names any name you like.
46. Select Compose Model and drag a Button to your form.

47. You should now see a button on your form with the default title *Action 1.*
48. In the **Configure Button** area click on the “…” in the **Action** field.

49. In the **Define Action** window select the **Submit** and select **Submit**.
50. Select Close.

51. You button now is configured with the action to **SUBMIT**. This is required when using Visual composer with guided procedures. Actions enable the guided procedure to know when the Visual Composer application has completed and to pass parameters back to the guided procedures context.

52. Rename the **Text** of the button to be **OK**.
53. Return to the design layout of by selecting the [Design] element at the bottom of the pane. You are now back in your design canvas.

54. Click on the Out link on the form. Drag your cursor.

55. Release to see the Context menu and select End Point.

56. Right-click on the line between the form and the endpoint. Select Configure from the context menu.
57. In the **Configure Data Map** select the “…” in the **Event name** field. Select **submit** as the event name. We are saying that when the submit button is selected, this indicates that the application has completed.

![Configure Data Map](image)

58. Select **Deploy**.

![Deploy](image)

59. In the **Deploy** window you should be able to see DC and **Runtime**. If you see it, skip to step 62.

However, in this diagram we do not see **Runtime**, which means we cannot deploy this application. If you see this view, perform the following steps.

![Deployment View](image)

60. To fix this select **Tools → Options** from the menu bar.
61. In the **Compiler** options select **Show runtime provider in Deploy task**.

62. Now you can **Compile**. Ensure **Web Dynpro HTML** is the selected **Runtime**. Select **Compile** then **Deploy**.

63. Once the deploy completes you can see your application. Select your application to test it.
64. You will receive a new browser window with your fields and an OK button. When you select the button you will get a blank screen.

65. Now we will add your Visual Composer application as a callable object for the final display. Return to your process in the Guided Procedures Design Time. Select your action AC_DisplayBP and select Create to create a new callable object.

66. Enter the following information for the callable object:

   **Name:** CO_DisplayBP
   
   **Description:** display the business partner
   
   **Type:** User Interface → WD4VC Application.

   Select Next.
67. Look for your Visual Composer application. You will see your development component and your application. In the diagram you're your recently created VC UI.

Select Next until you can select Finish. Optionally, notice the input parameters for the callable object.

68. Test your callable object by selecting the callable object, select the Test tab and Execute. You will see your Visual Composer application.

Select Save all.

CONGRATULATIONS! You have now built the process. Take a quick breather, then we have to do the parameter mapping for the ApproveBlock, then process, do some more setup, then soon we’ll be testing!!!
Step 4: Parameter Mapping and Role Assignment for Approval Block

Now we will do the parameter mapping and the role assignment for the approval block. Please take your time during the section and ensure you understand the parameter mapping.

1. Parameter mapping in the **ApproveBlock** will take several steps. Once this section is complete, ensure you look at each mapping and understand why we did the mapping like we did.

   There is a lot of mapping that needs to happen in this block. We will break it into three sections:
   - Information that will be input to the web service to create the business partner
   - Return values from the web service and return from the data in the ERP system
   - Setting up the correct user for the approval step

2. We will start with the mapping the information to the creation of the business partner. The fields from the **AC_ReviewBPIInformation** must be mapped to the **AC_ApproveBusinessPartner** for approval. They must also be mapped to the input parameters for the **AC_CreateBPinERP**.
3. Select your ApproveBlock and select the Parameters tab.

4. We will begin with mapping the FirstName. Select the FirstName from AC_ReviewBPInformation. Press the <CTRL> key and select the FirstName from AC_ApproveBusinessPartner.

5. Scroll down until you see the Input Parameters for AC_CreateBPinERP. Open the InputParameters. Press the <CTRL> key and select FIRSTNAME.

7. Name this group **GRP_Approve_BP_First_Input**. This denotes it is a group from the approve block, the first name of the business partner used as input on the web service.

   Select Create.

8. Your group should look like the diagram.

9. Repeat the steps for the **LastName** creating the group **GRP_Approve_BP_Last_Input**.
10. Repeat the steps for the **SearchBy** field, mapping **SearchBy** to **Search1** for the ERP business partner creation. Name the group **GRP_Approve_BP_SearchBy_Input**.

11. Repeat the steps for the **AlternativeSearch** field, mapping **AlternativeSearch** to **Search2** for the ERP business partner creation. Name the group **GRP_Approve_BP_Search2_Input**.

Select **Save all**.

12. Now we will work on the mapping of the output of the business partner creation step.

We need to map the business partner number returned from **AC_CreateBPinERP** to the input of **AC_GetBPDetails**. We also need to map the business partner number to the **AC_DisplayBP**.

We also want to map the return of our **AC_GetBPDetails** to the final display. This way we ensure we are displaying business partner information from the ERP system. In the **AC_DisplayBP** we also need to map the **Project Title** and **Project Description**, but those fields are not part of the **ApproveBlock**, so we will map them later.

In the end, you will have the following groups:

- **GRP_Approve_BusinessPartnerNumber_Output**
- **GRP_Approve_First_Output**
- **GRP_Approve_Last_Output**
- **GRP_Approve_SearchBy_Output**
- **GRP_Approve_Search2_Output**
13. Repeat the steps for the `SearchBy` field, mapping `SearchBy` to `Search1` for the ERP business partner creation. Name the group `GRP_Approve_BP_SearchBy_Input`.

14. Repeat the steps for the `AlternativeSearch` field, mapping `AlternativeSearch` to `Search2` for the ERP business partner creation. Name the group `GRP_Approve_BP_Search2_Input`.

Select `Save all`.

15. Now we will work on the mapping of the output of the business partner creation step.

We need to map the business partner number returned from `AC_CreateBPinERP` to the input of `AC_GetBPDetails`. We also need to map the business partner number to the `AC_DisplayBP`.

We also want to map the return of our `AC_GetBPDetails` to the final display. This way we ensure we are display business partner information from the ERP system. In the `AC_DisplayBP` we also need to map the Project Title and Project Description, but those fields are not part of the ApproveBlock, so we will map them later.

In the end, you will have the following groups:

- `GRP_Approve_BusinessPartnerNumber_Output`
- `GRP_Approve_First_Output`
- `GRP_Approve_Last_Output`
- `GRP_Approve_SearchBy_Ouptut`
- `GRP_Approve_Search2_Output`
16. We will start with the mapping of the business partner number. Select the Return Parameter of AC_CreateBPinERP and select the BUSINESSPARTNER. Press the <CTRL> key and select BUSINESSPARTNER from the Input Parameters of AC_GetBPDetails.

17. Scroll down to the Input Parameters of AC_DisplayBP. Press the <CTRL> key and select BusinessPartnerNumber.

18. Select Group and name the group GRP_Approve_BusinessPartnerNumber_Output. Your group should look like the diagram.

19. Now we will map the return from the AC_GetBPDetails to our final AC_DisplayBP. To do this select the Return Parameters from AC_GetBPDetails. The data is returned to us in structures. Select the structure CENTRALDATAPERSON and select FIRSTNAME.
20. Scroll down to the Input Parameters for AC_DisplayBP and select FirstName.

21. Select Group and name the group GRP_Approve_First_Output.

22. Repeat the steps for the LastName. Name the group GRP_Approve_Last_Output.

23. We will now do the mapping for the SearchBy and SecondarySearchBy fields. The SearchBy will be mapped to the SEARCHTERM1 return from ERP. The SecondarySearchBy will map to the SEARCHTERM2 return from ERP.

Select the AC_GetBPDetails Return Parameters and use the structure CENTRALDATA. Select SEARCHTERM1.
24. Press <CTRL> and select the SEARCHBY field from the AC_DisplayBP Input_Parameters. Select Group.

25. Name the group GRP_Approve_SearchBy_Output.

26. Repeat the steps for the SecondarySearchBy / SEARCHTERM2 and name the group GRP_Approve_Search2_Output. Select Save all.

27. There is one final set of mapping we need to do. We will now map the result of AC_PickUser to AC_AssignUserToRole. To do this ensure you are in the Parameters tab for your ApproveBlock.

28. Select UniqueID from AC_PickUser. From the AC_AssignUserToRole drill into User List → UserItem and select UserIdentifier. Select Group and name the group GRP_Approve_SelectedUser.
29. For the selected user to work, we need to update the role assignments. Select the Roles tab for the ApproveBlock.

30. Select the roles Processor of AC_AssignUserToRole and Processor of AC_ApproveBusinessPartner. We will consolidate these roles which will mean that the approval step will be done by the assigned user.
31. Enter **Approver** in the **Consolidate To** field and select **Go**.

32. Your role list will now look like the diagram.

33. Return to the **Parameters** tab so we can update the list of **Exposed In** parameters. Many parameters default to **Exposed In**. However, we only need parameters marked as **Exposed In** where parameter value should be passed in from the **MainBlock**.
34. Remember in the **MainBlock** business partner information was entered. We want that information passed to the **ApproveBlock**. Ensure the following parameters are **Exposed In**:

- `GRP_Approve_BP_First_Input`
- `GRP_Approve_BP_Last_Input`
- `GRP_Approve_BP_SearchBy_Input`

*InputParameters from AC_DisplayBP*

All other parameters should not be **Exposed In**. Please toggle all other parameters so they are not **Exposed In**.

Select **Save all**.
Step 5: Parameter Mapping for the Process

We will now update the parameter mapping for the process. We must ensure we have the correct data flow between the blocks.

1. Within our process we need information to flow between the blocks. The FirstName, LastName, and SearchBy fields entered in the **MainBlock** need to be passed to the **ApproveBlock**. The **Project Title** and **Project Description** from the **ProjectBlock** need to be passed to the **ApproveBlock**.

You will have the following groups:

- GRP_Process_BP_First
- GRP_Process_BP_Last
- GRP_Process_BP_SearchBy
- GRP_Process_ProjectTitle
- GRP_Process_ProjectDescription
2. Select your process and select the Parameters tab.

3. Select the FirstName from the MainBlock.

4. Press <CTRL> and select GRP_Approve_BP_First_Input.
   Select Group.

5. Name the group GRP_Process_BP_First.
6. Your group should look like the diagram.

7. Repeat the steps mapping LastName from the **MainBlock** to GRP_Approve_BP_Last_Input from the **ApproveBlock**. Your group name should be **GRP_Process_BP_Last**.

8. Repeat the steps mapping GRP_Approve_BP_SearchBy_Input to GRP_Main_SearchBy. Call the group **GRP_Process_BP_SearchBy**.

9. Now we will map the project title and description from the **ProjectBlock** to the **ApproveBlock**. Select the **GRP_Prj_Title**.

10. Select **Input_Parameters** from the **ApproveBlock** and select **Project Title**.

11. Select **Group** and name the group **GRP_Process_ProjectTitle**. Your group should look like the diagram.
12. Repeat the steps for the project description. Name the group **GRP_Process_ProjectDescription**. This should map the **GRP_Pri_Description** from the **ProjectBlock** to the **ProjectDescription** from the **Input_Parameters** for the **ApproveBlock**.

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
<th>Input_Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRP_Process_ProjectDescription</td>
<td>* ProjectDescription</td>
<td>ApproveBlock</td>
</tr>
<tr>
<td></td>
<td>* GRP_Pri_Description</td>
<td>ProjectBlock</td>
</tr>
</tbody>
</table>

13. We do not need any information passed to our process when it starts, so ensure no parameters are marked as **Exposed In**.

Select **Save all**.
Step 6: Role Assignment and Final Setup for the Process

We will do role assignment and some final setup such as ensuring our process follows the correct steps in the correct flow.

1. Select your process and select the *Roles* tab.

2. Currently we have several roles for our process. Other than the approval steps, the remaining steps will be done by the same person.
3. Select the Approver row and change the Role Type to Runtime Defined. This way the Approver will be determined at runtime from the pick user step.

<table>
<thead>
<tr>
<th>Role</th>
<th>Type</th>
<th>Items</th>
<th>Role Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin</td>
<td>Extended</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Override</td>
<td>Extended</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Owner</td>
<td>Extended</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>BusinessUser</td>
<td>Group 4</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Processor of AC_Register</td>
<td>Group 1</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Processor of AC_Display</td>
<td>Group 1</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Processor of AC_ReviewPriceInformation</td>
<td>Group 1</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Processor of AC_OrderDetails</td>
<td>Group 1</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Processor of AC_CreatePrintEP</td>
<td>Group 1</td>
<td>Initiator</td>
<td>▼</td>
</tr>
<tr>
<td>Approver</td>
<td>Group 1</td>
<td>Runtime Defined</td>
<td>▼</td>
</tr>
</tbody>
</table>

4. Select the other roles for the ApproveBlock, select the Business User role, and enter BusinessUser in the field Consolidate To. Select Go.

5. Your role assignment should now look like the diagram.
Select Save all.

6. We will now adjust our process to ensure it follows the correct path. First open up the AC_CheckIfEMEA action in the MainBlock.

There are two Result States: isEMEA, isNOTEMEA. Currently the output of the result states is not set.
7. If the value is isEMEA we want to jump to the ProjectBlock.

For the isEMEA select Other… from the drop down. Then select your ProjectBlock and the AC_EnterProject then press Select.

8. The result should look like the diagram.

9. For the isNotEMEA result select Other…
10. Select the **ApproveBlock** and the **AC_ReviewBPInformation**. Press **Select**.

11. The result should look like the diagram.

12. **OPTIONAL:** Now select **AC_ApproveBusinessPartner** in your **ApproveBlock**. Notice the **Result States**. There is one for approval, and one for rejection of the business partner.

13. **OPTIONAL:** If the business partner is rejected we want to terminate the block. Update the result state for the rejection to **Terminal**.

14. Select **AC_ApproveProject** in the **ProjectBlock**. If the project is rejected, update the result state to return to **AC_EnterProject**. Also update the approve result state to **AC_CreateProject**.

Select **Save all**.
15. Now it’s time to **Activate** your process and get ready to test!!

**Step 7: Test**

In this first test we will test the **MainBlock** going to the **ApproveBlock**.

1. Select your process and select the tab **Instantiation**. Select **Start Process Automatically**, **Generate Instantiate URL** and **Open Instantiate URL**.

2. A new browser will open and your process will start. Notice that you are in the **MainBlock**.
3. Enter a first name, last name, and search by of your course. But ensure the search by field is not EMEA. Select Complete.

4. Notice you are now in the ApproveBlock. The first name, last name, and search by fields have successfully been passed between blocks.

5. Enter information in the alternative search field and select Complete.
6. You are now in the pick user step. The user you enter here should be the user that does the approve step.

7. For this first test select enter your user ID. Enter the appropriate user (here: CE251) and select Go. Once the user information is displayed select Choose.
8. You are now in the visual approval step. The fields should be populated with the correct information you previously entered. Enter a brief comment and select Approve.

9. After selecting approve you may get something like what is shown in the diagram. There are several things here. One is the message: *The next activity is not yet available...* This is because the next step is the web service call to create the business partner and to get the details, so you are waiting for the service calls. Additionally you see the messages *Recipient data is missing* and *The system failed to send an e-mail notification.* This is because there is no email user ID assigned to user **CE251**. If you do not get these messages, then the email from the visual approval step was sent to whatever email account is in the user record.
10. Select *Refresh*.

11. You should now be in the final step which displays information from the ERP system. You will see a business partner number, first name, last name, search terms. The project title and description fields are blank because we did not execute that block.

Select OK.

**CONGRATULATIONS** on your first successful test!!!

12. OPTIONAL: Update the user record CE251 with your personal email account. Select *Personalize*. It is close to the *Log off* selection to log off the portal.
13. Select User Profile then select Modify and update the email address for the user account. Select Save when you are done.
Optional step 8: Use a Service from the SDN Service repository

In this exercise you will call an external web service from the SDN repository.

1. Locate the web service on sdn.
   b) Select Services Registry for the ES Workplace.
   c) In the window that appears select ERP Foundation → Business Partner Data Management → Customer. Then select Go.
   d) Select the service definition "CustomerSimpleByNameAndAddressQueryResponse_In".
   e) Click on the WSDL definition. A new browser window will launch and ask for a user name and password. Enter Demo for the user ID and welcome for the password.
   f) Highlight the WSDL address in the browser and select <CTRL>C to copy the definition to your clipboard. Do not copy the entire definition, just up to the wsdl1.1 entry:

2. Test the web service in the web services navigator.
   a) Go to http://localhost:50100/wsnavigator. Login as CE251/teched07.
   c) When asked for a UserID and password enter Demo and welcome.
   d) Test the web service by selecting the Operation: CustomerSimpleByNameAndAddressQueryResponse_In.
   e) This will execute the web service. This web service finds customer by certain criteria. We will find customers who are in the city Chicago. Look for the field CustomerAddressCityName. Enter Chicago. Select Execute.
   f) To drill into the response and look for Customer → BasicData → Common → Name. You will see one of the customers is CMR Corporation.

3. Look at the logical destination created for this services in the NetWeaver Administrator.
   a) Go to http://localhost:50100/nwa.
   b) Select Configuration Management → Infrastructure → Web Services Configuration.
   c) Select WS Destinations. Select the entry for SDN_FASTTRACK_CUSTOMER_SERVICE. Notice the WSDL URL and click on the Security tab to see the Demo user credentials provided.
4. Create a callable object that uses this web service.

   a) Go to http://localhost:50100/irj. Login as your user ID.
   b) Navigate to Guided Procedures → Design Time.
   c) Select your appropriate folder. Create a new folder called Optional.
   d) Select your Optional folder and create a callable object, entering the following information:
       
       Name: CO_SDNService
       Description: CO_SDNService
       Type: Service → Web Service
   e) Select the logical destination SDN_FASTTRACK_CUSTOMER_SERVICE.
   f) Select Search.
   g) Select CustomerSimpleByNameAndAddressQueryResponse_In. Select Accept.
   h) Select the operation and notice the input parameters. Select SDN_FASTTRACK_CUSTOMER_SERVICE for the Logical Destination Port.
   i) Select Next until you can select Finish and Open.
   j) Test your callable object, providing Chicago as the city.

Optional step 9: Ad Hoc Steps

In this optional step you will add an ad hoc step. This means when the user is processing a step, they have the option to do something in addition to that step. It will display in an “Options” area at runtime. In this task you will create an action and a callable object, then you will link this to your AC_InputBPInformation step.

Please note, this exercise will provide a description of what needs to be done, but there will not be a picture for each step. © This exercise includes the following major steps:

- Create an action and callable object called AC_RevenueReport and CO_RevenueReport.
- Update your process, AC_InputBPInformation, to use this AC_RevenueReport on an optional basis.
- Update the runtime view to show ad hoc actions.
- Test


The callable object should call a web page: http://localhost::50100/revenue.

   a) Navigate to your CE251_GrpXX folder in the Guided Procedures → Design Time.
   b) Select to create a new action. Name the action AC_Revenue. Add a callable object to the action using the following information:

       Name: CO_RevenueReport
       Description: execute a revenue report
       Type: User Interaction → Web page
   c) Select Next. Enter the URL: http://localhost:50100/revenue. Before entering this URL, open up a new browser, test to ensure it works correctly.
   d) Select Next until you can select Finish. Activate your action and callable object.
2. Update the AC_InputBPInformation step in your process to have an ad hoc action.
   a) Open up your process, select the Main Block and the action AC_InputBPInformation. Select Change.
   b) Select the tab Ad Hoc Items. Select Add.
   c) Select your action AC_RevenueReport as the Adhoc action.
   d) Select Save all.

3. Add the ad hoc view to the runtime view and test.
   a) Select your process and select the Runtime Views tab.
   b) Select Add generic view.
   c) Select the view “Process Activities (allows to view and execute activities and access activity related information and start ad-hoc items).”
   d) Select Add.
   e) Select the view and select Move Up so it is the first view listed in the tab Runtime Views.
   f) Select Save all to save your entire guided procedure. Activate.
   g) Test your guided procedure, notice the You can area when you are on the AC_InputBPInformation step. You can execute the URL, or select someone else to do the work.