Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

Applicable Releases:
SAP Fiori Approve Requests 1.0
SAP NetWeaver Gateway 2.0 SP 07
SAP Fiori Launchpad

Version 3.1
June 2014
SAP “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way.
<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The first release version. With Gateway 2.0 SP06 and Approve Request SP01.</td>
</tr>
<tr>
<td>2</td>
<td>With Gateway 2.0 SP07 and Approve Request SP03.</td>
</tr>
<tr>
<td>3</td>
<td>Simplified. Removed check steps.</td>
</tr>
<tr>
<td>3.1</td>
<td>Step 6 ABAP code is updated with escaping value</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example Text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚨</td>
<td>Caution</td>
</tr>
<tr>
<td>🛡️</td>
<td>Note or Important</td>
</tr>
<tr>
<td>⬤</td>
<td>Example</td>
</tr>
<tr>
<td>📜</td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
Table of Contents

Business scenario ........................................................................................................................................... 1
  Sample scenario ........................................................................................................................................ 1
  Prerequisites .............................................................................................................................................. 1

1. Step 1. Get workflow template information [ERP] .................................................................................. 2
  1.1 Find the workflow template ID ....................................................................................................... 2
  1.2 Find the approval step ID and step type ............................................................................................ 4
  1.3 Find the task ID .................................................................................................................................. 6
  1.4 Find the Business Object and method for Activity step type .............................................................. 6
  1.5 Find the outcomes from the step ......................................................................................................... 7

  2.1 Define scenario .................................................................................................................................... 9
  2.2 Assign consumer type ....................................................................................................................... 10
  2.3 Assign role ........................................................................................................................................ 10
  2.4 Assign task ......................................................................................................................................... 10

  3.1 Start the launchpad designer (system admin action) ........................................................................ 12
  3.2 Check the Target Mapping tile Click on the target mapping tile and make sure the data matches the list below ........................................................................................................... 13
  3.3 Add a new “App Launcher – Dynamic” tile ....................................................................................... 13
  3.4 Add app to Launchpad (End user action) ............................................................................................ 15

4. Step 4. Configure the task decision [ERP] ............................................................................................. 18
  4.1 Define workflow template and the step .............................................................................................. 18
  4.2 Assign the step keys .......................................................................................................................... 18

5. Step 5. Implement BADI for decision outcomes [ERP] ........................................................................ 20
  5.1 Create BAdI Configuration and Class for Decision Handling .......................................................... 20
  5.2 Implement BAdI Class for Decision Handling ................................................................................... 26

  6.1 Including business data by BADI (Optional but Recommended)[Gateway] .................................... 30

7. Test the scenario ....................................................................................................................................... 34
  7.1 Approve from SAP Fiori Approve Requests ...................................................................................... 34
Business scenario

With SAP Fiori Approve Requests, you can add workflow tasks to the Approve Requests. SAP delivers several standard approval scenarios but standard scenarios are not enough for your business. Now you are going to add additional custom workflow scenarios to SAP Fiori Approve Requests.

If making an approval/reject decision is simple, you can go live with the generic contents of Approve Requests. It is an easy and quick solution without big development effort.

If additional functions and UI screens are required, you can enhance the app or develop a custom app.

Sample scenario

In this exercise we will show you how to add “Notification of Absence” workflow scenario which is the demo workflow of SAP Business Workflow. It is available in all AS ABAP systems. This workflow is an Absence Approval scenario. It has an Activity step type which is bound to a BOR object/method. Your job will be to expose this to SAP Fiori Approve Requests app.

Implementation steps are 6 steps.

1. Get workflow template information
2. Define workflow scenarios
3. Add a tile to the Launchpad
4. Configure the task decision
5. Implement the BADI for task outcomes
6. Display application data

Prerequisites

SAP Fiori Approve Requests is used for illustration throughout this exercise.

- The workflow scenario is up and running in the SAP Business Suite system.
- SAP NetWeaver Gateway 2.0 SP07 or higher system and basic settings are configured.
- SAP Notes for SAP Business Workflow have been applied. The notes are listed in the SAP Fiori Installation and Configuration guide.
- One of the standard SAP Fiori approval apps are configured and running.
1. Step 1. Get workflow template information [ERP]

Before configuring the scenario, you should know the workflow template information.
Note: If you are in an SAP Education course, different workflow template ID and task ID are assigned to each user. Please get your assigned workflow template ID and task ID from your instructor and replace the WSXXXXXXXX and TSXXXXXXXX by your assigned ID.

Table: Workflow Information.

<table>
<thead>
<tr>
<th>No</th>
<th>Information</th>
<th>Your scenario</th>
<th>FORMABSENC sample in this doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Workflow template ID</td>
<td>WSXXXXXXXXX</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Approval step ID</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Step type</td>
<td>Activity</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Task ID</td>
<td>TSXXXXXXXXX</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Business Object</td>
<td>FORMABSENC</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Approval Method</td>
<td>APPROVE</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Outcomes</td>
<td>Container or Event</td>
<td>Container value</td>
</tr>
<tr>
<td></td>
<td>Approve</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Reject</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>Others</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.1 Find the workflow template ID

You are already using workflow scenarios in the production system. Find the workflow template ID that you are going to integrate with SAP Fiori Approve Requests.
Note: This step will show you where to go to obtain the required information however no activity will be performed here.

List workflow template last 365 days

1. Launch the SAP GUI (Start -> SAP Logon for Windows -> ZME System -> enter assigned credentials)
2. If you’re already in the GUI enter/N to return to the home screen in the ZME system then enter Transaction: SWI2_FREQ and choose enter from your keyboard
3. Select Last 365 days, uncheck Dialog and check (Sub-) Workflow then choose Execute

4. Select the Task Header and Sort in descending order

5. Workflow Templates are prefixed with 'WS', reference the workflow number for Z00 user (WS97100480).

   Remember this step is to show you where to find workflow templates and how to filter.

   NOTE: The WS30000015 Process Notification of absence is the sample workflow in this Exercise.
1.2 Find the approval step ID and step type

Display the workflow template

6. Enter Transaction: /NPFTC

7. Choose Workflow template in the task type drop down and manually enter your assigned task number (only the number as shown in the image below) in the task field. Then choose the glasses button to display

8. Choose Workflow Builder
9. Notice the approval step is 48. Choose this step by double clicking on it.

10. Check the step type (At the top of the screen above the tabs)

   Explanation of Step types (for our exercise we want Activity step type)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Step type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Activity" /></td>
<td>Activity</td>
<td>Function Module or Transaction will be called. Most of approval tasks are activity type.</td>
</tr>
<tr>
<td><img src="image" alt="User Decision" /></td>
<td>User Decision</td>
<td>User will select decision options manually</td>
</tr>
<tr>
<td><img src="image" alt="Web activity" /></td>
<td>Web activity</td>
<td>Web application integration</td>
</tr>
<tr>
<td><img src="image" alt="Form" /></td>
<td>Form</td>
<td>From integration</td>
</tr>
</tbody>
</table>

If the approval step type is Web activity or Form type, you need to adjust workflow template. Please contact workflow consultants to adjust.
1.3 Find the task ID

1. Note your approval task ID (it should match the task number that was assigned to you) and double click it.

![Workflow Builder - Display 'AF_process'](
![Image](image_url)
)

1.4 Find the Business Object and method for Activity step type

(Most of approval steps are activity type which calls Business Object method.)

2. Notice the Business Object is `FORMABSENC`
Method is `APPROVE`
1.5 Find the outcomes from the step

1. Select F3 from your keyboard to return to the previous Workflow Builder – Display screen.
2. Choose the Outcomes tab.
   Notice Approved and Rejected are listed as possible outcomes for the approve step.

3. Select the outcome Approved and Rejected by double clicking them and see the result value.
   The result value of Approved is “A”.
   The result value of Rejected is “R”.

4. Now you have all required workflow information.
   Table: Workflow Information.

<table>
<thead>
<tr>
<th>No</th>
<th>Information</th>
<th>Your scenario</th>
<th>FORMABSENC sample in this doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Workflow template ID</td>
<td>WS30000015</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Approval step ID</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>
### Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Step type</td>
</tr>
<tr>
<td>4</td>
<td>Task ID</td>
</tr>
<tr>
<td>5</td>
<td>Business Object</td>
</tr>
<tr>
<td>6</td>
<td>Approval Method</td>
</tr>
<tr>
<td>7</td>
<td>Outcomes</td>
</tr>
<tr>
<td></td>
<td>Approve</td>
</tr>
<tr>
<td></td>
<td>Reject</td>
</tr>
<tr>
<td></td>
<td>Others</td>
</tr>
</tbody>
</table>
2. Step 2. Define workflow scenarios [Gateway]

2.1 Define scenario

Set the workflow scenarios in the task gateway customizing in the SAP NetWeaver Gateway.

1. Enter Transaction: /NPRO -> SAP Reference IMG

   SAP NetWeaver -> Gateway Service Enablement -> Content -> Task Gateway Service -> Scenario Definition

   **Display IMG**

   - SAP Customizing Implementation Guide
     - Activate Business Functions
       - SAP NetWeaver
         - Gateway Service Enablement
           - Backend Data Channel
             - Content
               - Workflow Settings
                 - Task Gateway Service
                   - Context Service Settings
                     - Scenario Definition
                       - Registration
                         - General Settings
                           - Business Process Enablement
                             - Generic Channel
   - Gateway

2. Search for your entry – if it is not there enter the following information

   **Change View “Scenario Definition”: Overview**

<table>
<thead>
<tr>
<th>Field</th>
<th>value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario Identifier</td>
<td>Z##_ABSENCE</td>
<td>Technical ID</td>
</tr>
<tr>
<td>Scenario Display Name</td>
<td>Z## Absence Form</td>
<td>Task group in SAP Fiori Approve Requests</td>
</tr>
<tr>
<td>Scenario order</td>
<td>blank</td>
<td>blank</td>
</tr>
<tr>
<td>Technical Service Name</td>
<td>/IWPGW/TASKPROCESSING</td>
<td>Technical Name for generic Inbox service. Select by F4 help.</td>
</tr>
<tr>
<td>Version</td>
<td>1</td>
<td>Version of Technical service name.</td>
</tr>
<tr>
<td>Entity Set External Name</td>
<td>Task</td>
<td></td>
</tr>
<tr>
<td>Property External Name</td>
<td>TaskDefinitionID</td>
<td></td>
</tr>
<tr>
<td>Default Sort by Property</td>
<td>CreatedOn</td>
<td>Sort property</td>
</tr>
<tr>
<td>Mass Action</td>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

2.2 Assign consumer type
With your scenario highlighted choose Assign consumer Type to Scenario by double clicking it and notice the 3 available types

2.3 Assign role
If you want to enable the scenarios and consumer type to specific user groups, you can assign roles. The users who have assigned role get task box in the SAP Fiori Approve Requests. If there is no role assigned, all users will get the task box. This is an optional setting.

2.4 Assign task
You can assign multiple tasks to the Scenario. Choose Task Definition for Scenario by double clicking it.
NOTE: In all of these steps we just looked and didn’t make any changes. The point is to show you the different task assignments and how to assign them.
3. **Step 3. Add a tile to the Launchpad [Gateway]**

3.1 **Start the launchpad designer (system admin action).**

Launch Google Chrome and choose quick Link LPD Designer CUST

Launch Google Chrome and choose quick Link LPD Designer CUST

If you receive a security warning choose ‘proceed anyway’ and if prompted enter your login credentials and choose Log In

![Authentication Required]

The server http://2meidc08.wclt.sap.corp:55030 requires a username and password. The server says: SAP NetWeaver Application Server [2ME/800].

User Name: fiori-XX
Password: ********

Log In  Cancel

Select the SAP FND Transactional Applications (SAP_FND_TC_T_X1) catalog

![Select SAP FND Transactional Applications]

SAP FND Transactional Applications (SAP_FND_TC_T_X1)
3.2 Check the Target Mapping tile

Click on the target mapping tile and make sure the data matches the list below

Semantic Object: WorkflowTask.
Action: approveRequests
Launchpad Role: UIX01CA1
Launchpad Instance: TRANSACTIONAL
Application Alias: ApproveRequests

Once confirmed choose the back arrow

3.3 Add a new “App Launcher – Dynamic” tile

Choose the blank tile
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

Choose App Launcher – Dynamic

Choose this new tile and enter the information below:

**Title:** Enter the name of your workflow scenario, for example, Approve Absence Requests.

**Icon:** Choose whatever you like

**Service URL:**
- Go to Start -> Documents -> SAPX03_95
- Open the file named “Unit 4_Lesson10_Workflow_01.txt” and locate the URL
- Copy the URL
  
  /sap/opu/odata/IWPGW/TASKPROCESSING:mo/ScenarioCollection?$filter=key%20eq%20'ZXX_ABSENCE'

  **NOTE:** Change Z00 to match your number ZXX
- Paste the URL into the Link to Application space as shown below

**Semantic Object:** Enter WorkflowTask.

**Action:** Enter approveRequests.

**Parameters:** Enter scenarioid=<your scenario identifier> (Example: scenarioid=Z00_ABSENCE)

Choose Save
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

On the Confirmation Popup choose OK

3.4 Add app to Launchpad (End user action).

Make sure your user has the role SAP_FND_TCR_T_X1.
Open a new Google Chrome tab and choose the Launchpad quick link
Click on “Open Catalog”. Select the “Approve Absence Requests” app by clicking on the + sign.

Choose the back arrow to return to the application. The application will show.
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

Employee Fiori: Approve notification of absence

- Employee Fiori submitted a notification of absence on 14.01.2014.
- Make a decision about the request. If you reject the request, create an attachment explaining your reason for the rejection.

Created On: January 14, 2014
Description: Employee Fiori submitted a notification of absence on 14.01.2014. Make a decision about the request. If you reject the request, create an attachment explaining your reason for the rejection.
4. Step 4. Configure the task decision [ERP]

4.1 Define workflow template and the step

To expose the possible decision options of an Activity task type assigned to a BOR method with a Dialog we need to configure the decisions options.

1. Go to IMG and enter Transaction: SPRO (enter /NSPRO if you’re not on the homepage of the GUI)
   Choose SAP Reference IMG
   Content Navigate to SAP NetWeaver -> Gateway Service Enablement -> Content -> Workflow Settings and choose the execute button next to Maintain Task Names and Decision Options

2. In the list of workflows look for your Workflow ID. If it is not listed choose New Entries and enter the information as described in the table below. If it is listed choose it by selecting the gray box next to the Workflow ID.

<table>
<thead>
<tr>
<th>Field</th>
<th>value</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow ID</td>
<td>WS30000015</td>
<td>Workflow template ID</td>
</tr>
<tr>
<td>Step ID</td>
<td>48</td>
<td>Step ID</td>
</tr>
<tr>
<td>Icon MIME Repository</td>
<td>blank</td>
<td></td>
</tr>
<tr>
<td>Step Description</td>
<td>Approve absence</td>
<td>Task name in UI</td>
</tr>
</tbody>
</table>

4.2 Assign the step keys

The key order should be exactly the same as task definition outcomes. Choose Decision Keys by double clicking it.
Note: If the step type is User Decision type, you can skip this step 2. The first option is assigned to POSITIVE and the second option is assigned to NEGATIVE. We are just looking at the entries and there is nothing further to accomplish in this step.
5. Step 5. Implement BAdI for decision outcomes [ERP]

5.1 Create BAdI Configuration and Class for Decision Handling

To handle a decision that is made on the task we need to configure and implement a BAdI that will contain the logic for doing this. The BAdI class will need to contain the same logic for completing the task that is done in the assigned BOR object dialog method. These dialog methods are all slightly different so you need to take it on a case by case basis when implementing the decision BAdI. So understanding how the step works in the workflow and how the dialog method is implemented is key in this process.

1. Enter transaction /NSE18 and choose enter from your keyboard

2. The Enhancement Spot for getting Business Context data is /IWWRK/ES_WF_WI_BEFORE_UPD_IB. Enter that into the Enhancement Spot input field then choose the Display button

3. You should now see the following:

4. Drill into the BAdI Definition by choosing the dropdown arrow so you see the following:

5. Right click on the Implementations node and select Create BAdI Implementation in the context menu:
6. In the popup window click the create icon in the bottom right hand corner:

7. In the Create Enhancement Implementation enter the details as shown below then choose the green check mark.

8. If you receive the Create Object Directory Entry enter $TMP in the package then choose the Save button.
9. You will be navigated back to the Select or Create Enhancement Implementation popup window. Make sure your newly created Enhancement Implementation is selected as shown below and choose the green check mark.

![Select or Create Enhancement Implementation](image)

10. The Create BAdI Implementation popup window will now be displayed:

Enter in the details as shown below being sure to replace the XX with your assigned number then choose the green check mark

<table>
<thead>
<tr>
<th>BAdI Implementation:</th>
<th>ZXX_BADI_ABSENCE_FORM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>ZXX BADI Implementation Absence Form</td>
</tr>
<tr>
<td>Implementing Class</td>
<td>ZXX_IC_ABSENCE_FORM</td>
</tr>
</tbody>
</table>

![Create BAdI Implementation](image)

11. On the Create Object Directory Entry popup choose the save button
12. You should now see the following:

13. Go into Change mode by click the Display/Change button.
14. Drill into the BAdI implementation you just created by choosing the dropdown arrow and double clicking the Filter Val.

```
Enhancement Implementation  Z20_E1_ABSENCE_FORM

Properties | History | Technical Details | Enh. Implementation Elements
```

- BAdI Implementation: Z200_BADI_ABSENCE
- Description: Z200 BADI Implementations
- Default Implementation
- Example Implementation
- "Active" not switchable through Custom (IMG)

15. Choose the change mode button again and on the right hand side you will see the filter details view displayed then choose the Create Combination button.

```
Enhancement Implementation  Z210_E1_ABSENCE_FIORI Change

Properties | History | Technical Details | Enh. Implementation Elements
```

16. In the popup window select both WORKFLOW_ID and STEP_ID by holding down Shift and choosing the gray box next to each and choose the green check box.

```
Choose Filter

Available Filters

Filter
- WORKFLOW_ID
- STEP_ID

Filter Values

Filter Combinations

- Filter Combinations
- Filter
- Value 1

- Filter Values
- Filter
- Value 1
```

17. You should now see the following:
18. Double click on the ‘????’ in the Value 1 column for the STEP_ID row. You should see the following popup window:

![Image of filter popup window]

19. Enter 48 into the Value 1 input field (48 is the step ID for Approve absence notification), and set Comparator 1 to ‘=’. The filter should now look like so. Choose the green check mark.

![Image of filter with value 48]

20. Now set the WORKFLOW_ID filter choose the ??? Value 1 setting and enter ‘WSXXXXXXX’ (where the X’s are replaced with your assigned number) and Comparator 1 to ‘=’. Then choose the green check mark.

21. Your filters should now be set as shown below:

![Image of filters set for workflow scenario]

22. Activate your BAdI by clicking the Activate button.
23. In the Inactive Objects for FIORI-XX user choose the green check box

24. **Note**: you might see a warning telling you that the method that holds the actual implementation code has not been implemented…this will be done in the following section.

Congrats! You have created a BAdI that can be called to access that business context data of the Absence approval notification step.

### 5.2 Implement BAdI Class for Decision Handling

The BAdI class implementation for decision handling is going to be dependent on the workflow, the task, and the BOR Object Dialog method that the task is bound to. You will need to examine all of those elements to determine the necessary implementation. In any case you will be making use of the SAP_WAPI Business Workflow functions.

1. In your Enhancement Implementation switch to the Enh. Implementation Elements tab and double click Implementing Class node.
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

2. You will be displayed the following on the right hand side of the screen. Double click the method.

![Image](image1.png)

3. In the Create Method Implementation popup choose Yes.

![Image](image2.png)

4. Switch to change mode and add the follow code between the method and endmethod lines.

   - Go to Start -> Documents -> SAPX03_95
   - Open the file named “Unit 4_Lesson09_Workflow_01.txt” and locate the code
   - Copy the code (displayed below)
   - Paste in between Method and end method lines

   Please read the code and understand it!

```
DATA ls_obj TYPE swr_obj_2.
DATA lv_objtype TYPE swr_struct-object_typ.
DATA lv_objkey TYPE SWR_STRUCT-OBJECT_KEY.
DATA lv_retcode TYPE sy-subrc.
DATA lt_container TYPE TABLE OF swr_cont.
DATA ls_container_line TYPE swr_cont.
DATA FORMNUMBER TYPE SWXFORMABS-FORMNUMBER.
DATA ls_formabs TYPE swxformabs.

*Access the workflow data
CALL FUNCTION 'SAP_WAPI_GET_OBJECTS'
EXPORTING
  workitem_id = is_wi_details_wi_id
IMPORTING
```

Jun 2014
leading_object_2 = ls_object.

*Get the formnumber which is the key to the absence table
MOVE ls_object-instid TO formnumber.

*Select the details of the absence from the table SWXFORMABS
select single * from swxformabs
  into ls_formabs
  where formnumber = formnumber.

*Read the workflow's container data
CALL FUNCTION 'SAP_WAPI_READ_CONTAINER'
  EXPORTING
    workitem_id = ls_wi_details-wi_id
  IMPORTING
    return_code = lv_retcode
  TABLES
    simple_container = lt_container.

* Check which decision was selected and set the data
  * values appropriately
CASE iv_decision_key.
  WHEN 0001. "Approved"
    ls_container_line_value = 'A'.
    ls_formabs-procstate = 'A'.
  WHEN 0002. "Rejected"
    ls_container_line_value = 'R'.
    ls_formabs-procstate = 'R'.
ENDCASE.

_"_WI_RESULT is what the workflow keys off to determine
"which path to follow - Approve or Reject path
ls_container_line_element = ' _WI_RESULT'.

*Modify the workflow's container data - we are updating the row that
  "holds _WI_RESULT which will be in the second row of the table
MODIFY lt_container INDEX 2 FROM ls_container_line TRANSPORTING value.

"Write the updated container back to the workflow engine
CALL FUNCTION 'SAP_WAPI_WRITE_CONTAINER'
  EXPORTING
    WORKITEM_ID = ls_wi_details-wi_id
    DO_COMMIT = 'X'
  IMPORTING
    RETURN_CODE = lv_retcode
  TABLES
    SIMPLE_CONTAINER = lt_container.
IF lv_retcode NE 0.
  "Handle error
ENDIF.
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

5. Save and Activate your changes.

6. On the Inactive Objects for Fiori-XX popup choose the green check mark

6.1 Including business data by BADI (Optional but Recommended)[Gateway]

You can enhance the workitem description by implementing BADI. HTML format is supported in workitem description area.

The Enhancement spot name is /IWPGW/ES_TGW_TASK_DATA.
Please refer the section 5 for details steps of BADI implementation.

Please make sure you maintain the filter.
PROVIDER_ID = BWF
TASK_DEF_ID = your task ID. TS########

Method is MODIFY_TASK_DESCRIPTION.
- Go to Start -> Documents -> SAPX03_95
- Open the file named “Unit_4_Lesson09_Workflow_02.txt” and locate the code
- Copy the parameters (one at a time)
- Paste the code
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

. Please read the code and understand it!

The value should be escaped to avoid cross side script attacks.
Please look at help document.
val_escaped = escape( val = val format = cl_abap_format=>e_xss_ml ).
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

```abap
* SUBCONTAINER_BOR_OBJECTS = lt_subcontainer_bor_objects
SUBCONTAINER_ALL_OBJECTS = lt_subcontainer_all_objects.

if lv_returncode = 0.

*retrieve only the object key line
read table lt_subcontainer_all_objects with table key element = lc_element
into wa_subcontainer_all_objects.

if sy-subrc = 0.
lv_formnumber = wa_subcontainer_all_objects-value.
CALL FUNCTION 'SWX_FORMABS_READ' destination lc_backend
EXPORTING
  FORMNUMBER = lv_formnumber
IMPORTING
  FORMDATA = ls_formdata
RETURNCODE = lv_returncode.
if lv_returncode = 0.

CONCATENATE CV_DESCRIPTION '<div><p>You can display business data in HTML view by BADI.</p>' into CV_DESCRIPTION.
CONCATENATE CV_DESCRIPTION '<table border = "1">' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-formnumber format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Form Number</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-createdate format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Created</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-name format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Name</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-department format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Department</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-pernr format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Personel No.</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-costcenter format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>Cost center</td><td>' val_escaped '</td></tr>' into CV_DESCRIPTION.

val_escaped = escape( val = ls_formdata-firstday1 format = cl_abap_format=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '<tr><td>From</td><td>' val_escaped
```
Adding a custom workflow scenario to the SAP Fiori Approve Requests in 6 steps

```abap
'\</td></tr>\' \into\ CV_DESCRIPTION.
val_escaped \= escape( val \= ls_formdata-
lastday1 format \= cl_abap_format\=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '\<tr><td>To</td><td>' val_escaped '
</td></tr>\' \into\ CV_DESCRIPTION.
val_escaped \= escape( val \= ls_formdata-
holidaytp1 format \= cl_abap_format\=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '\<tr><td>Leave Type</td><td>' val_escaped '
</td></tr>\' \into\ CV_DESCRIPTION.
val_escaped \= escape( val \= ls_formdata-
reason format \= cl_abap_format\=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '\<tr><td>Reason</td><td>' val_escaped '
</td></tr>\' \into\ CV_DESCRIPTION.
val_escaped \= escape( val \= ls_formdata-
contact format \= cl_abap_format\=>e_xss_ml ).
CONCATENATE CV_DESCRIPTION '\<tr><td>Contact at</td><td>' val_escaped '
</td></tr>\' \into\ CV_DESCRIPTION.
CONCATENATE CV_DESCRIPTION '</table>' \into\ CV_DESCRIPTION.
endif.
endif.
endif.
endmethod.
```
7. **Test the scenario**

7.1 **Approve from SAP Fiori Approve Requests**

1. Launch the Fiori Launchpad by going to Google Chrome and choosing the Launchpad quick link. If prompted enter your logon credentials.

2. Select the Approve Absence Requests form tile we created in an earlier step.

3. Select the approve icon.

4. Make a comment and choose Submit.

5. You should receive a successful message and the workflow item is removed from the list.