**Build a Workflow from Scratch with SAP Cloud Platform Workflow Management**DEV163

Exercise 1 | Workflow  
Archana Shukla / SAP

Table of ContentS

[Overview 3](#_Toc52990122)

[MODEL WORKFLOW using business application studio 4](#_Toc52990123)

[build and deploy WORKFLOW 45](#_Toc52990124)

[execute and Monitor WORKFLOW 47](#_Toc52990125)

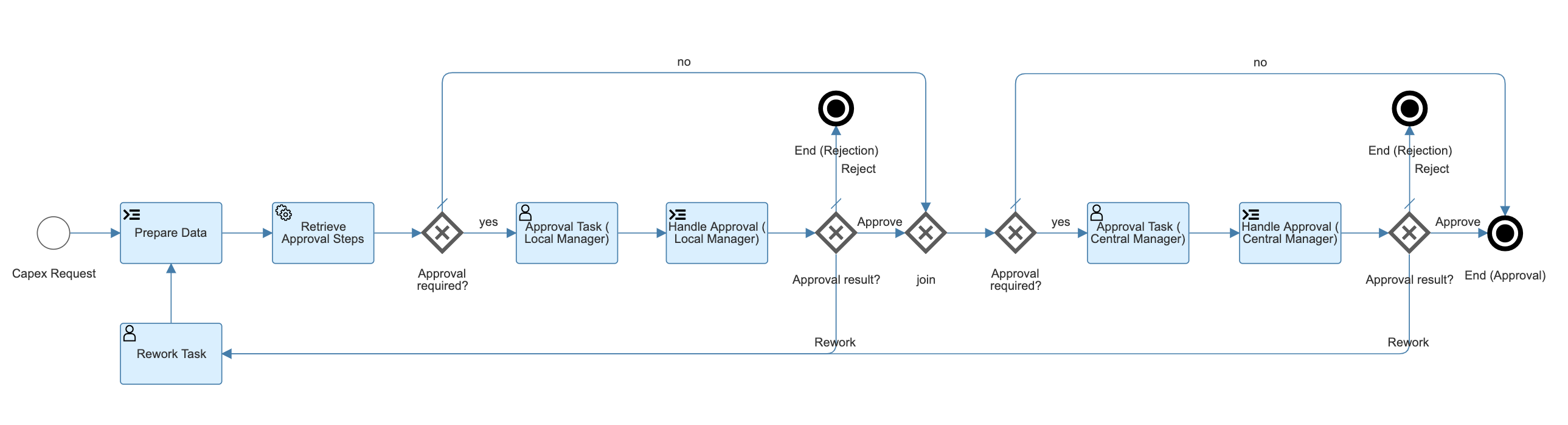
[Summary 53](#_Toc52990126)

# Overview

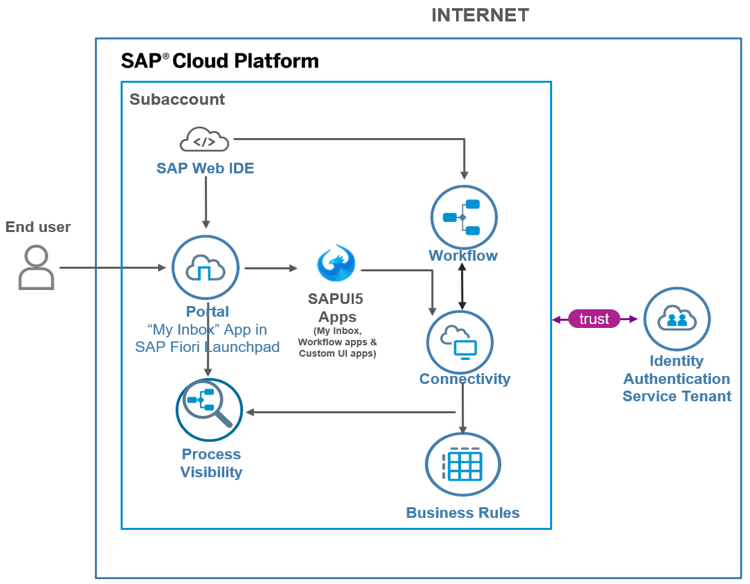
Estimated time: 90 minutes

Capital Expenditure process starts when any company receives a funding request for capital expenditures. These requests are then thoroughly reviewed, evaluated and finally approved or rejected based on the available budgets for the current period. Approval is not straight forward. There are many rules that are involved to determine how many levels of approvals will be there or who will approve it based on the investment details.

There are many challenges with capital expenditure process and companies are always looking for managing investment decisions in a transparent way. Automating capital expenditure approval process enables customer to accelerate investment decisions and to bring the agility into every step and decision such that the corporate policies are followed while taking the investment decisions.



The below solution diagram explains how customer can automate CAPEX scenario using SAP Cloud Platform.



# MODEL WORKFLOW using business application studio

| Explanation | Screenshot |
| --- | --- |
| 1. Logon to **SAP Cloud Platform Trial Home Page** with your trial username and password.   <https://cockpit.hanatrial.ondemand.com/cockpit/#/home/trial>   1. Choose **SAP Business Application Studio**. |  |
| Each SAP Business Application Studio Development Space is a virtual machine hosting your project files, securely separated from other spaces.   1. If you have no dev space, click the **Create Dev Space** and follow instructions in step 5. 2. Else select **PLAY** 46F1C2F18C37A68E to start the space if you see the status as **STOPPED**.   Note: Idle dev space will be shut down after some time. This is indicated by a yellow bar in the bottom of the screen with the text "Offline". In this case, return to this Dev Space selection screen from your bookmark and restart it. | imb__610 |
|
| 1. Enter **workshop** as the new **Dev Space name**. 2. Ensure **Application Type** is selected as **SAP Fiori**. 3. Select the extension **Workflow Management**. Note: This extension will enable the workflow modelling features. 4. Select **Create Dev Space**. |  |
| 1. Wait until the virtual machine providing this dev space is in status **RUNNING**. 2. Open the **SAP Business Application Studio** by clicking on the dev space name. | imb__612 |
| Note: SAP Business Application Studio is a web application that load the features of the extensions. There might be a short delay for all features to be operational when first accessed. | imb__613 |
| 1. From the menu, select **Terminal**. 2. Select **New Terminal** to open a Linux Terminal window. | imb__614 |
| 1. Enter **cd projects/** in the **Terminal** window to go into the folder where you are going to create the workflow module. Note: This project folder is already created. 2. Enter **yo** in the terminal window to start Yeoman.   Note: Yeoman is a tool for quickly setting up the basic structure of projects.   1. If asked, choose whether you want to enable telemetric for Yeoman. 2. Select **Basic Multitarget Application** by using the up/down arrow keys and press **Enter**. | imb__615 |
| 1. Enter **CapexProject-00** or any name of your choice as **Project name**. | imb__616 |
| 1. Change into the newly created directory by entering **cd CapexProject-00**. You can also use **TAB** for auto-completion, i.e. entering **cd Cap<TAB>** will be sufficient. 2. Enter **yo** in the window to start Yeoman again. 3. Select **Workflow Module** by using the up/down arrow keys and press **Enter**. | imb__617 |
| 1. Enter **Capex-00** as **Module Name**. 2. Enter **Capex\_00** as **Workflow Name**. 3. Enter **Capital Expenditure Approval** as **Workflow Description**. 4. Yeoman will warn you that there already is an **mta.yaml** file. Type **Y** to overwrite it. 5. You can close the **Terminal** by typing **exit** or clicking on the X in the window header. | imb__618 |
| 1. From the menu, select **File**. 2. Select **Open Workspace...**. | imb__619 |
| 1. Select the directory containing your module **CapexProject-00**. 2. Select **Open**. | imb__620 |
| 1. From the menu, select **File**.   Note that there is an **Auto Save** functionality (enabled by default). If you choose to disable it, remember to manually save your files. | imb__621 |
| 1. If it is not already open, select the **Explorer**fieldicon0000 from the sidebar. 2. Open the **Capex-00** module by selecting it. 3. Open the **workflows** directory by selecting it. 4. Open the **CAPEX\_00.workflow** by double-clicking on it.   Note: If the graphical workflow editor is not opened, or a textual (code) editor is opened instead, the workflow plugin did not load correctly. Refresh your browser window and try again after SAP Business Application Studio has fully loaded. | imb__622 |
| A new workflow with a default Start and End event is created.   * On the left side there is a toolbar for adding new flow elements. * On the right side is the properties pane for the currently selected element.  1. Click on the **Start Event**fieldicon to select it. This will show the quick action icons and the property pane will show the details of this element. 2. Enter **Capex Request** as the **Name** of the start event.   This will later be shown in the workflow *Execution Log* as well as in Process Visibility. | imb__623 |
| 1. Click on the tab **Details** of the start event. 2. Select **Configure Sample Context**. When starting a new workflow, an initial context (JSON data) can be given. With this option, an example context can be configured. 3. Click **Create File**. | imb__624 |
| * 1. In the popup, enter **CapexSampleStartPayload** as **File Name**.   2. Click **Create**. | imb__625 |
| A new file with the name **CapexSampleStartPayload.json**  is created in the project folder below **sample-data**. It is created with some default data as example. | imb__626 |
| * 1. Remove the example content and Copy-Paste the example context given in the file **CapexSampleStartPayload.json** into the editor.   2. Set the RequestId to CAPEX\_REQ\_XX\_001. Replace XX with the number assigned to you by the trainer.   3. Insert your name, email address and user id into the respective fields. Your **UserId** is **your trial user email ID**.   4. Close the file by clicking onfieldiconin the header bar. If shows a round circle instead this indicates that there are changes, which will be saved by closing. | imb__627 |
| * 1. Click the **Start event**95BC09DED32F0BA8 to select it.   2. Click on **Add Task**fieldicon0000.   3. Select the **Script Task**E8F1C061925352A3. | imb__629 |
| * 1. In the Script Task Properties, enter **Prepare Data** as **Task name.**   Note: We will configure the task later. | imb__630 |
| * 1. On the selected Script Task *'Prepare Data'*, click **Add Task**fieldicon in the speed buttons.   2. Select **Service Task**B156A06F7F30559E. | imb__631 |
| * 1. Enter **Retrieve Approval Steps** as **Name**.   Note: We will configure the task later. | imb__632 |
| * 1. On the selected *'Retrieve Approval Steps*' service task, click **Add Gateway**fieldicon0001.   2. Select **Exclusive Gateway**3428B7AFB6B881A1. | imb__633 |
| * 1. Enter **Approval required?** as gateway **Name**.   Note: We will configure the gateway later, once we added more outgoing branches. | imb__634 |
| * 1. Select **Add Task**fieldicon0000.   2. Select **User Task**9458B2BEAA1FB2. | imb__635 |
| Note that the **Task ID**is **usertask1**. This will be needed later.   * 1. Enter **Approval Task (Local Manager)** as **Task name**.   Note: IDs are automatically given to objects. If an object is deleted and re-created, it will get a new ID. | imb__636 |
| * 1. Select **Add Task**fieldicon0000.   2. Select **Script Task**98797F5EDE00888E. | imb__637 |
| * 1. Enter **Handle Approval (Local Manager)** as **Name**.   Note: We will configure the task later. | imb__638 |
| * 1. Click **Add Gateway**fieldicon0003.   2. Select **Exclusive Gateway**D889016873232787. | imb__639 |
| * 1. Enter **Approval result?** as **Name**. | imb__640 |
| * 1. Select **Add Gateway**fieldicon0001.   2. Select **Exclusive Gateway**4DD7724BEBF741A6. | imb__641 |
| * 1. Enter **join** as gateway **Name**.   Note: This gateway will be joining the branches from a previous gateway together. In most cases, such joining gateway are optional and implicit, when sequence flows point to the same element. However, when the next element is a gateway as well, this needs to be made explicit. | imb__642 |
| * 1. Select the **Exclusive Gateway**CF9F3E507EA005AE with the name **'Approval required?**'   2. Click on the **Sequence Flow Connector**fieldicon and start dragging it   3. Drop the **Sequence Flow Connector** onto the top corner of the **Join** gateway.   Note: Sequence Flow Connectors have Connection Dots at each end. Those can be placed either in the middle or on certain positions on the edge of Flow Elements. The Layout Assistant takes this into account when drawing the connector. Additionally, the lines can be dragged with the mouse. | imb__643 |
| * 1. Drag the **Connection Dot** fieldicon from the middle of the Exclusive Gateway with name **Approval required?** to its upper corner.   2. Drop on **the upper corner** of the icon. | imb__644 |
| Since there is now more than one Sequence Flow, each flow has a name and branch condition.   * 1. Enter **no** as **Name** for the newly created (upper) Sequence Flow.   2. Select **Default** to make this as the default branch. This will mean that if none of the condition will meet then this default route will be executed. | imb__645 |
| * 1. Select the **Sequence Flow**8C9322CDF67DAC8E between the gateway *'Approval required?*' and the user task *'Approval Task (Local Manager)*' | imb__646 |
| * 1. Enter **yes** as **Sequence Flow name**.   2. Enter **${context.approvalStepsResult.Result[0].Approvers.lm\_required}** as **Condition**.   warnung_word Be careful during copying. There are no spaces in the condition. Sometimes during copy, spaces are added automatically. In this case, they need to be removed.  Note: This expression uses data in the workflow context to evaluate whether this branch should be taken. This data was retrieved in previous step by a service call to Business Rules. | imb__647 |
| * 1. Select the script task **Prepare Data**. | imb__648 |
| * 1. In the properties, click **Create File**. | imb__649 |
| * 1. Enter **PrepareData** as **Name**.   2. Click on **Create**. | imb__650 |
| * 1. Delete all the default content and replace it with the script given in the file **PrepareData.js**   Note: This workflow calls a pre-build rule which you don't need to change now. In a business rules exercise, a custom rule will be created and then you can replace it with your rule service id.   * 1. Close the file by clicking on fieldicon0002. |  |
| * 1. Select the Service Task **Retrieve Approval Steps**. | imb__652 |
| * 1. Select the **DETAILS** tab.   2. Enter **BUSINESS\_RULES** as **Destination**.   3. Enter **/rest/v2/rule-services** as **Path**.   4. Enter HTTP Method as **POST**.   5. Enter **${context.rulesPayload}** as **Request Variable**.   6. Enter **${context.approvalStepsResult}** as **Response Variable**. |  |
| * 1. Select the User Task **Approval Task (Local Manager)**. | imb__654 |
| * 1. Select the **DETAILS** tab.   2. Enter **Approval for '${context.Title}' in your role as Local Manager** as **Subject**.   3. Enter **${context.approvalStepsResult.Result[0].Approvers.lm\_userid}, ${info.startedBy}** as **Recipients - Users**.   It means that this user task will be sent to the users retrieved from Business Rules and the workflow initiator. |  |
| * 1. Select the **USER INTERFACE** tab.   2. Select **Form** in the dropdown menu for **Type**.   3. Click **Create File**.   Note: Forms are a quick way to create simple task UIs. | imb__656 |
| * 1. In the popup, enter **CapexApproval\_00** as **Form name**.   2. Press **TAB** to accept the generated Id.   3. Enter **1** as **Revision**.   4. Click **Create**. | imb__657 |
| Note: With Workflow Forms, building Fiori form UIs becomes as easy as building workflows. The Forms Editor enables a declarative low-code approach to rapidly develop simple form UIs. Due to its full integration into Workflow a form can directly work on the values of the workflow context. | imb__658 |
| * 1. Click **Add Section**. | imb__659 |
| * 1. Enter **Investment Details** as **Title**.   Note: The ID field is automatically filled out based on the given title. | imb__660 |
| * 1. Select **Add Field**. | imb__661 |
| * 1. Enter **Request Id** as **Label**. The ID is then automatically derived from this.   2. Enter **${context.RequestId}** as **Context Path**.   3. An input field can also be marked as **Required**. Since the **Request Id** is already given and should not be editable, leave it unchecked. | imb__662 |
| * 1. Clicking on the **Type** field opens a dropdown list. Select the type as **String**. | imb__663 |
| * 1. Clicking on the **Mode** field opens a dropdown list. Select **Display-Only**. | imb__664 |
| * 1. In the same way, add the following fields with given Label, Type and Context Path: * Set the property of all of the fields as **Editable** and **Required**.     Label: **Title**  Type: String  Context Path: ${context.Title}    Label: **Type**  Type: String  Context Path: ${context.Investment.Type}    Label: **Country**  Type: String  Context Path: ${context.Investment.Country}    Label: **CAPEX**  Type: Float  Context Path: ${context.Investment.CAPEX}    Label: **ROI**  Type: Float  Context Path: ${context.Investment.ROI}    Label: **Total Cost**  Type: Float  Context Path: ${context.Investment.TotalCost}    Label: **Currency**  Type: String  Context path: ${context.Investment.Currency}    Label: **Description**  Type: String  Context Path: ${context.Investment.Description} | imb__665 |
| * 1. Select **Investment Details**​ section.   2. Select **Add Section**. | imb__666 |
| * 1. Enter **Requester Details** as **Title**. | imb__667 |
| * 1. Add the following fields.     Label: **Name**  Type: String  Mode: Display-Only  Control: Input  Context Path: ${context.Requester.Name}    Label: **Comment**  Type: String  Mode: Editable  Control: Text Area  Required: no  Context Path: ${context.Requester.Comment} | imb__668 |
| * 1. Select the **Requester Details** section. | imb__669 |
| * 1. Select **Add Section**. Name the new Section **History**. | imb__670 |
| * 1. Select **Add Collection**.   Note: Collections allow for an array of items to be displayed as table. | imb__671 |
| * 1. Enter **History** as **Title**.   2. Enter **${context.History}** as **Context Path**. | imb__672 |
| * 1. Add the following values for **Label** and **Context Path** and select Type **String** and mark them as **Display-Only**.     Label: **User**  Context Path: ${item.UserId}    Label: **Role**  Context Path: ${item.Role}    Label: **Action**  Context Path: ${item.Action}    Label: **Comment**  Context Path: ${item.Comment}    Note that inside a collection, the variable **item**refers to a single row. | imb__673 |
| * 1. Add a new Section with title **Approval**.   2. Add a new field and name it **Comment**.   3. Enter **${context.comment}** as **Context Path**   4. Change the **UI Control type** from *Input field* to **Text Area**. | imb__674 |
| * 1. Ensure your Form looks like this. | imb__675 |
| * 1. Select the **Decisions** tab.   2. Select **Add** to add a new decision option. | imb__676 |
| * 1. Enter **Approve** as **Text** of the decision button.   2. Select **Positive** as **Type** from the dropdown-list. | imb__677 |
| * 1. In the same way, add a decision option **Reject** with type **Negative** and one with text **Rework** of **Neutral** type. | imb__678 |
| * 1. **Close** the file by clicking fieldicon_168. | imb__679 |
| * 1. Select the script task **Handle Approval (Local Manager).** | imb__680 |
| * 1. In the **Script Task Properties**, select **Create File**. | imb__681 |
| * 1. Enter **HandleApprovalLocalManager** as Filename   2. Select **Create**. | imb__682 |
| * 1. **Replace** the default content with the script from **HandleApprovalLocalManager.js** in the exercises folder.   2. **Close** the file by clicking fieldicon.   Note: This script refers to the Task ID (usertask1) multiple times. This is the id of the User Task "*Approval Task (Local Manager)*" which was mentioned earlier. It is automatically assigned to each task. If your ID is different, it needs to be replaced here. Please note only replace if your user task has a different id. | imb__683 |
| * 1. From the left sidebar, select **Add Event**.   2. Select the **Terminate End Event** E538B5D611DF39AB. | imb__684 |
| * 1. Place the **End Event** above the exclusive gateway named **Approval result?**   Note: The yellow guiding lines help you position it nicely. | imb__685 |
| * 1. Enter **End (Rejection)** as **Name**.   2. Ensure that **Terminate** is checked. | imb__686 |
| * 1. Select the exclusive gateway with name **Approval result?**   2. Drag the sequence flow connector fieldicon onto the end event with name **End (Rejection)** | imb__687 |
| * 1. Enter **Reject** as **Name**.   2. Select the **checkbox** for **Default** condition. | imb__688 |
| * 1. Select the sequence flow connector between the gateways named **Approval result?** and **join**.   Note: Once there is more than one outgoing sequence flow from an Exclusive Gateway, all of them can be named. Exactly one must have the 'default', and all others customized trigger conditions based on the workflow context. | imb__689 |
| * 1. Enter **Approve** as **Name**.   2. Enter **${context.decision == "approve"}** as **Condition**. | imb__690 |
| * 1. From the left sidebar, select **Add Task**EE191BDA30CA94B7.   2. Select **User Task**3C24A0ADFE39EC99. | imb__691 |
| * 1. Place the **User Task** below the script task **Prepare Data**.   Note: The yellow guiding lines help you position it nicely. | imb__692 |
| * 1. Verify that the **ID** is usertask2. If it differs, the references in the script file **Prepare Data.js** needs to be adapted accordingly.   2. Enter **Rework Task** as **Name**. | imb__693 |
| * 1. Select the **DETAILS** tab.   2. Select the **Priority** as **High**.   3. Enter **Rework for '${context.Title}'** as **Subject**.   4. Enter **${info.startedBy}** as Recipients - User.   5. Select **Configure Due Date**.   6. Enter **2** as **Duration** and **Days** from the dropdown list **Unit of Time**, to mark the task as being due 2 days after it started. |  |
| * 1. Select the **USER INTERFACE** tab.   2. Select **Form** as **Type**.   3. Click **Select** to choose an existing form. | imb__695 |
| * 1. Select the previously created form **CapexApproval\_00**   2. Ensure that the **Revision** is 1.   3. Select **OK**. | imb__696 |
| * 1. Select the Exclusive Gateway **Approval result?** and drag an additional **Sequence Flow Connector**... | imb__697 |
| ...onto the User Task **Rework Task**. Position it onto the highlighted middle dot. | imb__698 |
| * 1. Click on the **Connection Dot**4095BE05D78AAA2of the new Sequence Flow and drag it from the middle of the Exclusive Gateway... onto the lower edge of the Exclusive Gateway. | imb__700 |
| * 1. Ensure your outgoing Sequence Flow Connectors look like this. | imb__701 |
| * 1. Enter **Rework** as **Name**.   2. Enter **${context.decision == "rework"}** as **Condition**. | imb__702 |
| * 1. Select User Task **Rework Task**   2. Select **Sequence Flow Connector** and drag it... | imb__703 |
| ...onto the Script Task **Prepare Data**. | imb__704 |
| Ensure your workflow looks like this. | imb__705 |
| * 1. Select the exclusive gateway with name **join**.   2. Select **Add Gateway** fieldicon.   3. Select the **Exclusive Gateway** optionfieldicon0000.   4. Name it as **Approval required?** | imb__706 |
| * 1. Select **Add Task**fieldicon.   2. Select **User Task**6311423FCF688291.   3. Enter **Approval Task (Central Manager)** as **Name**. | imb__707 |
| * 1. Select the **Details** tab.   2. Enter **Approval for '${context.Title}' in your role as Central Manager** as **Subject**.   3. Enter **${context.approvalStepsResult.Result[0].Approvers.cm\_userid}, ${info.startedBy}** as **Recipients - Users**.   Note: During development, the user tasks will be sent to the process initiator. Later, the user Id will be retrieved from SAP Cloud Platform Business Rules. |  |
| * 1. Select the **User Interface** tab.   2. Select **Form** as **Type**.   3. Use the **Select** button to choose the previously created form **capexapproval\_00**. | imb__709 |
| * 1. Select **Add Task** on the central manager approval.   2. Select the **Script Task**. | imb__710 |
| * 1. Enter **Handle Approval (Central Manager)** as **Name**.   2. Select **Create File** to create a new file. Name it **HandleApprovalCentralManager** and paste the content of the file with the same name from the exercise folder. You can then close it again.   Note: This script refers to the **Task ID** (usertask3) multiple times. This is the id of the User Task **Approval Task (Central Manager)** which was mentioned earlier. It is automatically assigned to each task. If your ID is different, then it needs to be replaced here.    Note: To access the task decision the ID of the User Task is used. The difference in the two **Handle Approval** scripts are which task they refer to, and the 'Role' stored in the history element of workflow context. | imb__711 |
| * 1. Select **Add Gateway**fieldicon.   2. Select the **Exclusive Gateway**9600F070B454D5A6.   3. Name it as **Approval result?** | imb__712 |
| * 1. Click **Approval result?** and Select **Add Event**.   2. Select the **Terminate End Event. Name** the event as **End (Approval)**. |  |
| * 1. On the previous exclusive gateway with name **Approval required?** add a new **Sequence Flow Connector** and drag it onto the newly added **Terminate End Event**.   Note: To make it look nice, move the blue connection dots on both sides to the upper edge of the icons and optionally drag the line upwards further upwards. |  |
| * 1. Enter **no** as **Name**.   2. Select it as **Default**. | imb__715 |
| * 1. Select the Sequence Flow Connector fieldicon. between **Approval required?** and **Approval Task (Central Manager)**. | imb__716 |
| * 1. Enter **Yes** as **Name**.   2. Enter **${context.approvalStepsResult.Result[0].Approvers.cm\_required}** as **Condition**.     Note: Be careful during copying. There are no spaces in the condition. Sometimes during copy, spaces are added automatically. In this case, then remove them.  Note: how this accesses the field *cm\_required*, for "*approval required from Central Manager*", whereas for the first approval, the field was called *lm\_required*, for the *Local Manager*. | imb__717 |
| * 1. Create a new **Terminating End Event** and place it above the **Approval result?** gateway.   Name it **End (Rejection)**.   * 1. Drag a new Sequence Flow Connection from **Approval result?** onto the **End (Rejection)** event. Name it **Reject** and select **Default** as condition. |  |
| * 1. Select the Sequence Flow Connection from **Approval result?** onto the **End (Approval)**   2. Name it **Approve** and enter $**{context.decision == "approve"}** as condition. |  |
| * 1. Drag a new Sequence Flow Connection from **Approval result?** onto the **Rework Task**.   2. Adjust it that it starts directly downward by selecting the starting Connection Dot and move it to the lower edge of the Exclusive Gateway. |  |
| * 1. Name it as **Rework** and enter **${context.decision == "rework"}** as condition. |  |
| * 1. Click anywhere on an empty space in the workflow editor to open **Workflow Properties**.   2. In **Workflow Properties**, enter **Subject** as **CAPEX Approval for '${context.Title}'**   3. Enter **Business Key** as **${context.RequestId}**   4. Select the **ATTRIBUTES** tab. |  |
| Note: You can drag the vertical divider line of the **Workflow Properties** Pane to enlarge it.  Now we will add attributes from the workflow context that will be exposed to be seen in Process Visibility scenario (in exercise 3 later) | imb__728 |
| * 1. Select **Add**.   2. Enter **Currency** as **ID**.   3. Enter **Currency** as **Label**.   4. Enter **${context.Investment.Currency}** as **Value**. | imb__729 |
| * 1. In the same way, add the following fields:     **ID:** TotalCost  **Label:** Total Cost  **Value:** ${context.Investment.TotalCost}    **ID:** CAPEX  **Label:** CAPEX  **Value:** ${context.Investment.CAPEX}    **ID:** ROI  **Label:** ROI  **Value:** ${context.Investment.ROI}    **ID:** Country  **Label:** Country  **Value:** ${context.Investment.Country}    **ID:** Type  **Label:** Type  **Value:** ${context.Investment.Type}    **ID:** Title  **Label:** Title  **Value:** ${context.Title}    **ID:** RequestId  **Label:** Request Id  **Value:** ${context.RequestId}  Note: Currently only String type is supported. | imb__730 |
| Your workflow should finally look like this. |  |
| Note: If you did not use **Auto Save**, ensure that all files are saved. | imb__732 |

# build and deploy WORKFLOW

| Explanation | Screenshot |
| --- | --- |
| * 1. Double-click the file **mta.yaml** to open it.   Note: The default MTA will create a reference to a new workflow service instance. When you ran the Booster in the setup the workflow service instance was already created so change the resource to the existing instance name.   * 1. Change the resource name to **wm\_workflow**.   2. Change the resource type to **org.cloudfoundry.existing-service**.   3. Change the resource reference to **wm\_workflow**.   4. Close the tab and ensure the file is saved. |  |
| * 1. To build the project, right-click on **mta.yaml** to open the context menu.   2. Select **Build MTA**. | imb__734 |
|
| * 1. Ensure that the **Build MTA** task exits with code 0, indicating the build is completed. | imb__735 |
| * 1. Ensure that a cloud tenant has already been configured. It will be shown in the bottom left corner on a blue bar. If it is not configured yet, then click on the blue bar and follow the wizard to enter the details. | imb__736 |
| * 1. Select the newly created directory **mta\_archives** to open it.   2. Right-click on the created file **CapexProject-00\_0.0.1.mtar** to open the context menu.   3. Select **Deploy MTA Archive**. | imb__737 |
| * 1. Ensure that the **Deploy MTA Archive** task exits with code 0 indicating that the deployment has finished.   Note: You can also click on the 9AA339FC0D8EC0B1 icon in the lower right corner to toggle display of the console panels. | imb__738 |
| Note: Your deployment console should look similar to this with the "**Process finished**" in the second-to-last line, indicating success.  If there are any errors then fix the error, build and deploy the workflow project again. | imb__739 |

# execute and Monitor WORKFLOW

| Explanation | Screenshot |
| --- | --- |
| 1. Open the Workflow Management Launchpad 2. **Log On** with your trial username and password. 3. Select the tile **Monitor Workflows - Workflow Definitions**. |  |
|  |  |
| Note: **Monitor Workflows** application has Master-Detail layout where you see all the workflows deployed on left. Ensure that the workflow you just deployed - **CAPEX\_00** - is existing.   1. Select the workflow. | imb__742 |
| 1. Select **Start New Instance**.   Note: This way of starting a workflow is mainly used for testing. Workflows can be started by end users using a Start Form or a custom UI5 application, and by other systems using the [Workflow API](https://api.sap.com/api/SAP_CP_Workflow_CF/resource). | imb__743 |
| 1. Modify the **RequestId** in the JSON data to be unique: use CAPEX\_REQ\_00\_YYY where YYY is a different number each time you start a workflow, e.g. counting up from 001. 2. Also adapt the following fields in the JSON element **Requester**:  * **Name** can be anything * **Email** should be your trial user email ID * **UserId** is your trial logon user id  1. Select **Start New Instance**. | imb__744 |
| A popup confirms that the workflow instance has been started.   1. Select **Show Instances** to switch to the **Monitor Workflows - Workflow Instances** application where you can view all the running, completed, suspended workflow instances and their details. | imb__745 |
| Note: When coming from **Show Instances**, a **Filter** is automatically added to the specific **Workflow Definition**.   1. Select the newly started workflow instance and look at its details. 2. Select **SAP icon** to go back to the Home Page of the Workflow Management Launchpad. | imb__746 |
| 1. Select the tile for **My Inbox**. It should show at least one new task. |  |
| 1. Ensure that in the list view the new task "**Approval for...**" is selected. | imb__748 |
| 1. Scroll down to the **Approval** section. 2. Enter a **Comment** of your choice. 3. Click **Approve**. | imb__749 |
| 1. In this exercise, the user task in the workflow has been configured to send all tasks in your inbox. To see the second approval task, select **Refresh**fieldicon. | imb__750 |
| Note: You can now see the task for the Central Manager. | imb__751 |
| Note: The **History** table is filled with the comment entered in the previous task.   1. Enter again a **Comment** of your choice. 2. Click **Approve**. | imb__752 |
| 1. Select **SAP icon** to go back to the Workflow Management Home Page. | imb__753 |
| 1. Select the **Monitor Workflows - Workflow Instances** tile. |  |
| 1. Be default, **Monitor Instances** application only shows Running, Suspended and Erroneous workflow instance. To see the completed workflow, click on the line showing the currently active **Filters** or select the **Filter**D28599C12748D5BB icon at the bottom of the screen. | imb__755 |
| 1. In the popup, select **Status**. | imb__756 |
| 1. Select **Completed** to also show already completed workflow instances. 2. Select **OK**to activate the changed filter settings. | imb__757 |
| 26. Ensure the correct instance is selected.  27. Look at the details of the completed workflow instance.    Congratulations! You completed the Workflow exercise.  In this exercise, you learnt how to:   * Model workflow from SAP Business Application Studio. * Build & deploy workflow * Start workflow instance from Monitor Definitions * Track the workflow instances from Monitor Instances * Process your work items from My Inbox | imb__758 |

# Summary

You have completed the exercise!

You are now able to:

1. Create a new workflow in **Business Application Studio**.
2. Add & configure **service task**, **script task**, **user task** and **gateways** in the workflow model.
3. **Build** and **deploy** workflow from Business Application Studio
4. Test the workflow using **Monitor Workflow | Workflow Definition** application.
5. Track and monitor the workflow using **Monitor Workflow | Workflow Instances** application.
6. View user tasks and take action from **My Inbox** application.