Creating a Workflow Process Definition Template

About the Process Definition Template

Each tenant in SAS Customer Intelligence 360 has a planning item approval template and a sample content production workflow template. Custom workflow templates can be defined during the requirement definition stage. This documentation is available to any employee who works with customers and defines the workflow templates. The configurations of the template enable the Admin user to associate the templates with the SAS Customer Intelligence 360 items to create a workflow process. The template is configured in the Flowable Modeler and is an outline of a process definition. The process definition can be modified after it is in SAS Customer Intelligence 360. These process definitions can change after users start using them. When a change is required, the workflows, which have already started processing using the old definition, should continue processing according to the old definition. Any new workflows should start processing using the new template definition.

Prerequisites

Download the Flowable Modeler. In this documentation, we are using the 6.3.0 version of the modeler.

Create a Process Definition Template in Flowable Modeler

1 Launch the Flowable Modeler and click **Kickstart App**.

2 Click **Create Process**. Enter a unique Model Name and a unique Model Key. The Model Key is the process identifier.

3 By default, the start event is provided. In the section below the canvas, enter the ID and Name for the start event.

4 Define the sequence by adding the arrows. To add the arrows, click the user task and drag ↗ to the task. In the section below the canvas, enter the ID and Name for the sequence flow.

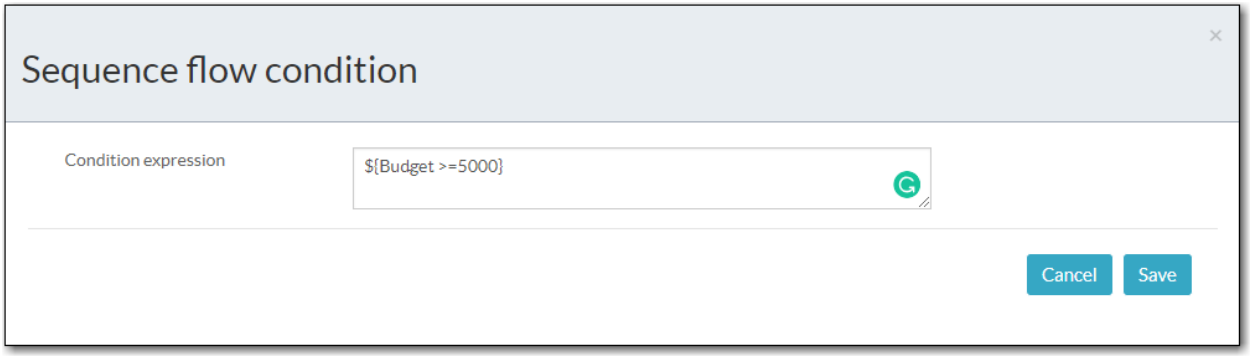
5 In a process, constructs such as user tasks, conditional sequence flows, and gateways are supported.

- You can create user tasks. For more information see "Create User Tasks" on page 4.
- You can create conditional sequence flows. For example, if the budget for Task-01 is greater than or equal to USD 5000, continue to Task-02; if the budget is less than USD 5000, continue to Task-03. To add a condition, click the arrow and enter the ID and Name in the section below the canvas.
  - Click **Flow condition**. In the **Condition expression** field, enter *${Budget>=5000}* for the arrow for Task-02 and enter *${Budget<5000}* for the arrow for Task-03.

    Do not use the words that are reserved in Java for an expression language (EL) in your variable names. For more information, see Oracle reserved words.

    Note: Except for "_", no special characters are allowed in variable names.

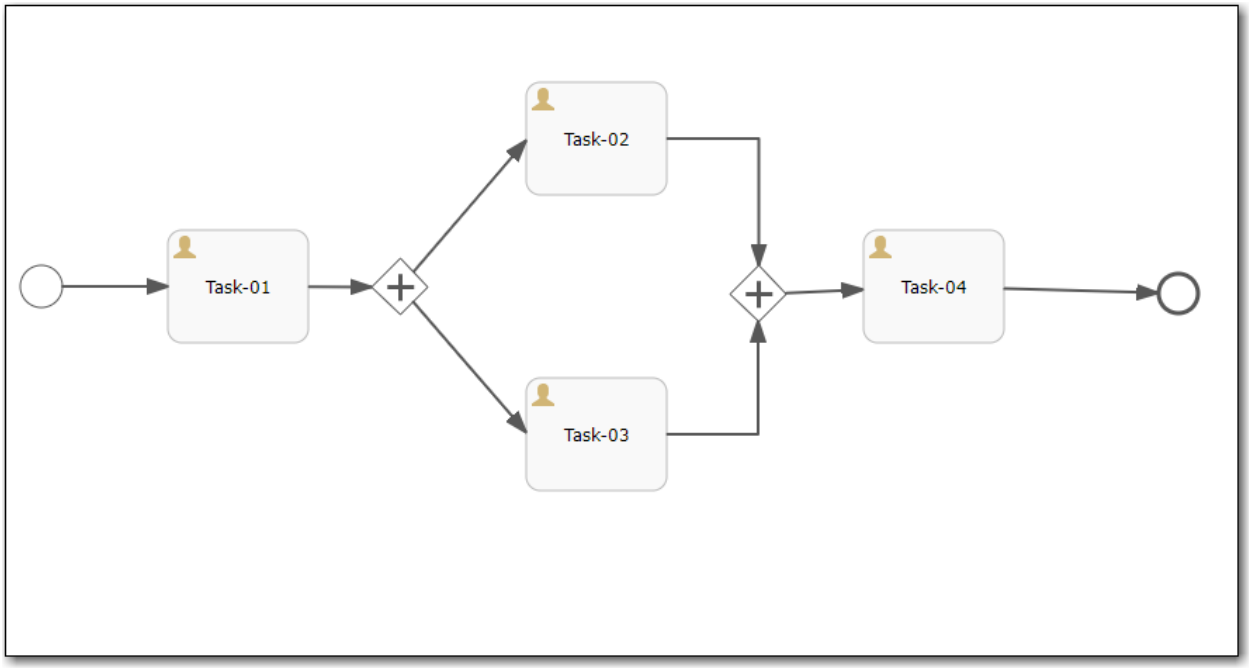
Figure 1 *Condition Expression*



- Create a gateway to control the flow of the process. The gateway defines how the process executes. To add a gateway, click the user task and select ◈ Next, enter a unique ID. From the gateway, drag an arrow to the next task and set the condition. For more information, see Step 4 on page 2.

  Example of Gateways:

Figure 2 Example



In the preceding example, Task-02 and Task-03 are executed in parallel. After a task is completed, the process stops at the gateway. After the second parallel task is completed, only one instance of Task-04 is created and initiated.

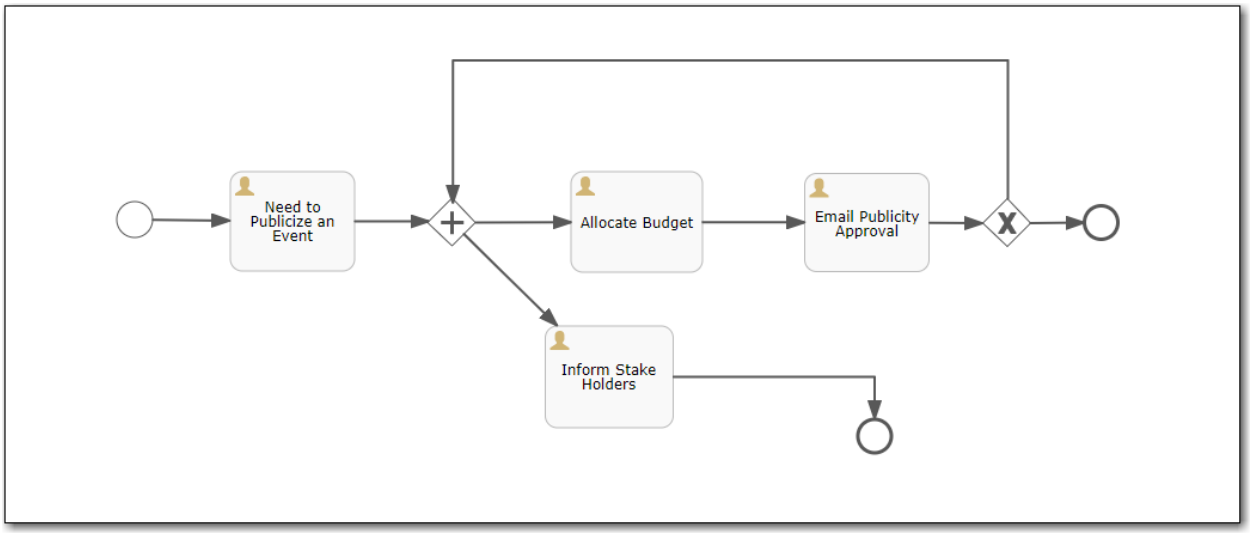
For more information about types of gateways, see "Supported Gateways" on page 6.

6 After you define the process, select ○ to end the workflow. Create the end event by navigating to **End Events** or by clicking the last user task and selecting ○.

Note: A task in a workflow can have only one active instance at a time. Cases where multiple instances of the task can be created, which results in more than one active instance of the task, are not supported. For an example, see Figure 4.

- The workflow is activated by initiating the Need to Publicize an Event task.
- After the Need to Publicize an Event task is completed, the Allocate Budget and Inform Stake Holders tasks are initiated.
- After the Allocate Budget and Inform Stake Holders tasks are completed, the Email Publicity Approval task is initiated.
- After the Email Publicity Approval task is completed, the Allocate Budget and Inform Stake Holders tasks are initiated.
- Initiation of the Allocate Budget task is supported because the task is completed, but initiation of the Inform Stake Holders task is not supported because when one instance of the task is active, another instance of the task cannot be created.

Figure 3 *Workflow with a Cyclic Loop*



Create User Tasks

You can create user tasks in either of these ways:

- Navigate to **Activities** and drag **User task** onto the canvas, or click the start event, and then click .
- In the section below the canvas, enter the ID and Name and retain the default values for the remaining properties. For a multi-instance task, enter values for these properties:

Table 1 *Property Values*

Property	Description
Multi-Instance type	Determines whether this task is performed multiple times and how the task is performed. Here are the possible values: ■ None: The task is performed once. ■ Parallel: The task is performed by multiple contributors simultaneously. ■ Sequential: The task is performed by multiple contributors one after the other.
Collection (Multi-instance)	(Used with the Multi-Instance type.) The name of a process variable, which is a collection. For each item in the collection, an instance of this task is created. For example, the value could be *parallelApprovers*. The value should be different from the Element variable and should be unique across tasks.

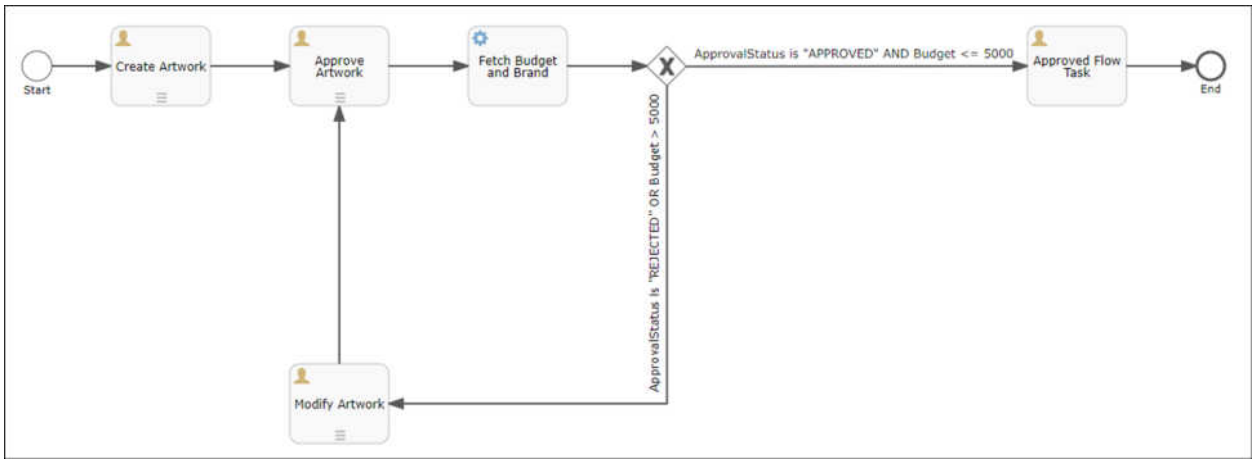
Property	Description
Element variable (Multi-instance)	The name of a process variable that contains the current value of the collection in each instance of a task. The value can be anything; there are no restrictions. For example, the value could be *parallelApprover*.
Assignments	Configures who the task is assigned to. For example, the value could be *${parallelApprover}*, which is the Element variable.

Note: The values for collection, element variable, and assignments properties must not include any spaces. Problems might occur when the templates are used in Workflow instances, resulting in errors.

Create Service Tasks

You can create a workflow template that includes a service task for external integration.

Figure 4 *Workflow for Service Task*



You can create service tasks in the following way:

- Navigate to **Activities** and drag a **Service task** onto the canvas.
- In the section below the canvas, enter the ID and Name. Enter values for the following properties and retain the remaining properties without any change:

Table 2 *Property Values*

Property	Description
Triggerable	The property determines whether a service task is triggerable. When the property is set to true, the service task executes and waits for a trigger event to continue to the next task in the workflow. The value of the property must always be set to true.
DelegateExpression	Allows you to specify an expression that resolves to an object that implements the TaskListener interface. The value of the property must always be specified as *${workflowServiceTaskDelegate}*.

Supported Gateways

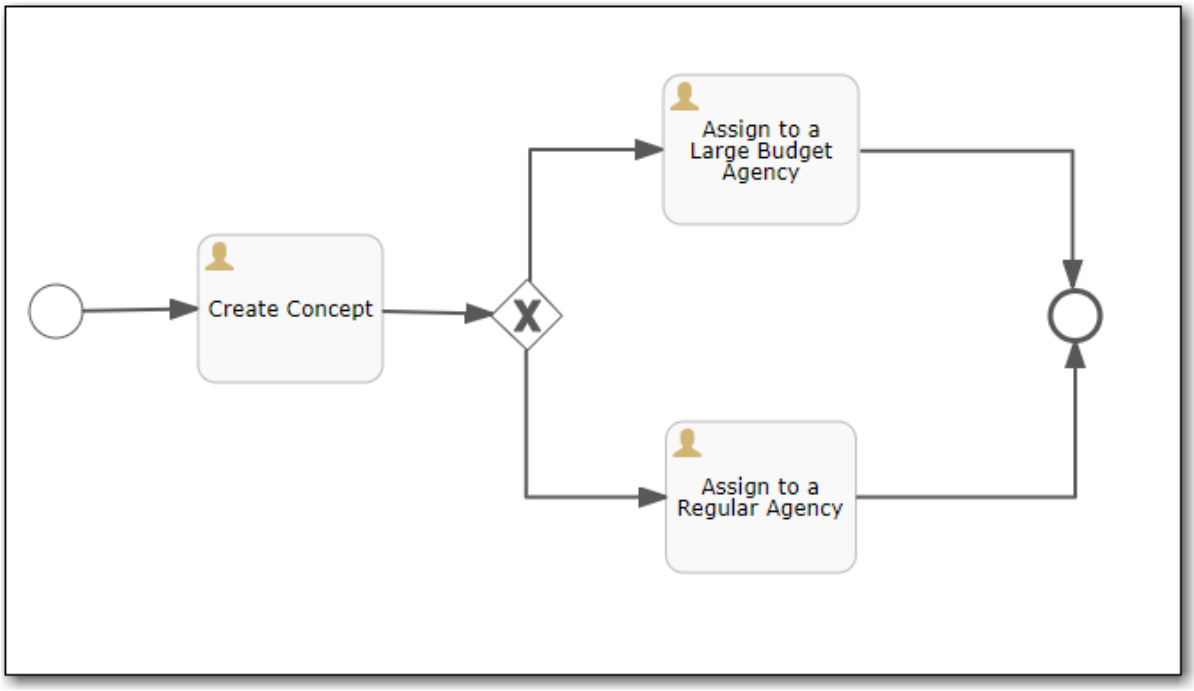
A gateway is used to control the flow of execution. By default, a gateway is represented as a diamond shape with an X inside  . In SAS Customer Intelligence 360, these gateways are supported:

- Exclusive Gateway: An exclusive gateway, which is also called the XOR gateway, is used to model a decision in the process. When the execution arrives at this gateway, all outgoing sequence flows are evaluated. The sequence flow, which a condition evaluates to true, is selected for continuing the process. An exclusive gateway is represented as a diamond shape with an X inside and refers to the XOR semantics.

  Note:

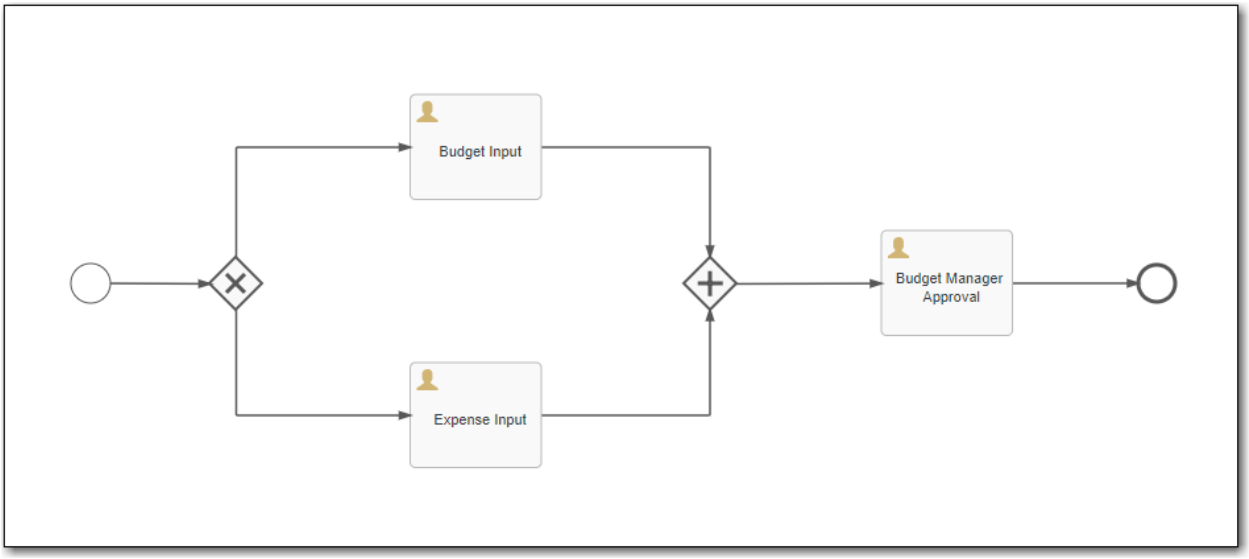
  If no condition is set for an outgoing flow, conceptually it is evaluated as true.

Figure 5 *Example of a Template with Exclusive Gateway*



- Parallel Gateway: Gateways can also be used to model concurrency in a process. The most straightforward gateway to introduce concurrency in a process model is the parallel gateway. This gateway allows the flow to fork into multiple paths of execution or join multiple incoming paths of execution. A parallel gateway is represented as a diamond shape with a plus symbol (+) inside and refers to the AND semantics.

Figure 6 *Example of a Template with Parallel Gateway*



- Inclusive Gateway: The inclusive gateway is a combination of exclusive and parallel gateways. Like an exclusive gateway, you can define conditions on outgoing sequence flows and the inclusive gateway will evaluate the conditions and the sequence flows. Like a parallel gateway, the inclusive gateway allows more than one sequence flow. An inclusive gateway is represented as a diamond shape with a circle inside .

Figure 7 *Example of a Template with Inclusive Gateway*