Contents

Chapter 1 / Introduction to SAS Workflow Manager .............................................. 1
What Is Business Process Management? ......................................................... 1
Why Use SAS SAS Workflow Manager? ......................................................... 3
SAS Workflow Manager on SAS Viya ............................................................ 3
The Workflow Definition Lifecycle ............................................................... 3
Sign In to SAS Workflow Manager .............................................................. 4

Chapter 2 / Creating and Editing Definitions .................................................... 7
About Workflow Definitions ........................................................................... 8
Create and Activate a Workflow Definition .................................................. 8
Edit Definition Properties ........................................................................... 10
Edit Element Properties ............................................................................. 11
Specify a Client Identifier ........................................................................... 11
Add Sequence Flows .................................................................................... 12
Add Sequence Flow Conditions ................................................................... 13
Add Annotations and Associations ............................................................... 14
Manage Permissions for a Definition ............................................................ 15
Import a Definition ....................................................................................... 16
Export a Definition ....................................................................................... 16
Create and Manage Data Objects ................................................................. 17
Creating and Managing Tags ....................................................................... 20
Manage Versions of Workflow Definitions .................................................. 21
Activate a Version ........................................................................................ 22
Delete Definitions ........................................................................................ 23

Chapter 3 / Working with User Tasks .............................................................. 25
About User Tasks .......................................................................................... 25
Add a User Task ........................................................................................... 26
Managing Participants .................................................................................. 26
Managing Prompts ....................................................................................... 33

Chapter 4 / Working with Service Tasks ......................................................... 35
About Service Tasks ....................................................................................... 35
About Service Task Transient Variables ...................................................... 36
Add a Service Task ....................................................................................... 36
The Invoke REST Web Service Action ........................................................ 37
Manage Output Data Objects for REST Web Service Tasks ....................... 38
Invoke REST Web Service Action Example .............................................. 39
The Invoke Web Service Action ................................................................... 40
Manage Output Data Objects for SOAP Web Service Tasks ....................... 40
Invoke Web Service Action Example ......................................................... 41
The Send Email Action ............................................................................... 42
The Send Email Action Example ................................................................. 42
The Send Notification Action ..................................................................... 43
The Send Notification Action Example ....................................................... 43
The Send Group Notification Action ............................................................ 44
The Send Group Notification Action Example ........................................... 45
The Invoke Job Execution Action ................................................................. 46
The Invoke Job Execution Action Example ............................................... 47
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Working with Subprocesses and Call Activities</td>
<td>51</td>
</tr>
<tr>
<td>6</td>
<td>Using Timers</td>
<td>55</td>
</tr>
<tr>
<td>7</td>
<td>Using Messages</td>
<td>59</td>
</tr>
<tr>
<td>8</td>
<td>Using Gateways</td>
<td>63</td>
</tr>
<tr>
<td>9</td>
<td>Localizing Workflow Text</td>
<td>67</td>
</tr>
<tr>
<td>10</td>
<td>Understanding Workflow Patterns</td>
<td>71</td>
</tr>
</tbody>
</table>
What Is Business Process Management?

Organizations of all sizes are forced to deal with a changing business environment. The changing marketplace, technological advances, and shifting customer priorities create challenges that businesses must overcome every day. All successful organizations need efficient processes to convert their competencies and resources into value for their customers. Success requires a delicate balance between establishing efficient, repeatable processes, and maintaining the agility to adjust or completely replace these processes to fit current conditions.

Common challenges include the following:

- people acting in concert
  The actions of good managers, along with prior training and experience, largely determine how effectively members of a group can work together to bring about an aggregate result. Yet all of these factors take time to develop, and might never fully develop if the pace of change is high. A well-designed process management system can help by orchestrating—by way of notifications, reminders, delivery of resources, and tracking—the work of many individuals involved in a business process. It can also automate much of the start up and cleanup in each individual activity. For example, it can automate finding the right forms and locating relevant policies, or automate forwarding to the next person in the process.

- interleaving automation
  Not all processes can be usefully automated, but even partially automated processes are more efficient than completely manual processes. A well-designed process management system can help identify where automation can have the highest impact, along with an operational framework for deploying and managing automated processes.
performance analysis and optimization

High-level summary results can indicate problems, but detailed analysis is required in order to pinpoint and fix bottlenecks and inefficiencies in operations. A properly implemented process management system can collect detailed metrics on the actual performance of key processes in real time. These metrics give management a concrete basis for making decisions about how and when to make improvements.

Business process management (BPM) is a disciplined approach focused on aligning all aspects of an organization on fulfilling the needs of its clients. It emphasizes integrating technology into the business process such that the process itself drives the business goals, decoupled from the underlying systems and applications. Specifically, BPM emphasizes how the work is done within an organization, in contrast to what a product does.

BPM can also be used to understand relationships between processes and relationships within the organization and across organizational boundaries. The analysis of those relationships, when included in a process model, enable sophisticated, horizontal reporting and analysis.

Critical success factors for BPM include the following:

- understanding the current state business process and client needs
- applying governance and standards based on business policies and practices
- using metric and key performance indicator (KPI) definitions that support measurable business goals

More specifically, a business process is a collection of activities designed to produce a specific output for a particular objective, possibly involving both human and system interactions. Essentially, a process is an ordered sequence of work activities defined with respect to time and place, with a beginning, an end, and clearly defined inputs and outputs: a structure for action.

Initially, BPM focused on the automation of business processes. It has evolved to integrate manual processes in which human interaction takes place in series or parallel with the use of technology. For example, when individual steps in a basic workflow require human intuition or judgment, these steps are assigned to members within the organization. Consequently, the difference between workflow and BPM is not distinct. Generally, workflow management is considered to be a subset of BPM that emphasizes static routing and administration of human tasks. In contrast, a business process might include a combination of automated and manual activities with dynamic routing based on embedded business logic. Today, many products include varying aspects of customization and control, but both approaches emphasize the elimination of bottlenecks, minimization of redundancies, and improved operational efficiency.

In short, workflow systems can be thought of as a type of operating system for the enterprise. The function of this system is to orchestrate and track work, whether automated or carried out by people. In the same way that databases capture what an organization consumes and produces, workflow systems encapsulate how the organization operates.
Why Use SAS SAS Workflow Manager?

Integration and interoperability of applications and data has improved with the emergence of the web, middleware technologies, enterprise application integration (EAI) efforts, and adherence to software standards like Java, J2EE, and XML. However, businesses in many industries manage the integration of applications, data, and people by adopting business process management (BPM) practices. Popular workflow features include the following:

- systematic routing of tasks requiring manual intervention
- automated triggering of basic actions and alerts

Process management systems offer the option to define, automate, audit, and refine business operations by leveraging the web, middleware, and standards to more efficiently and effectively manage by process. SAS Workflow Manager provides the tools to rapidly integrate fundamental workflow management into business operations and business offerings based on SAS solutions and products.

SAS Workflow Manager on SAS Viya

SAS Workflow Manager 2.3 integrates with the platform workflow service, which is based on an open-source engine that complies with the Business Process Model and Notation (BPMN) 2.0 standard. It supports industry standards for visual representation and behavior, and it provides support for established workflow patterns. Specifically, SAS Workflow Manager supports workflows that are compliant with the BPMN 2.0 standard. SAS Workflow is efficient and scalable, and it integrates with other standard business process implementations.

The Workflow Definition Lifecycle

To develop and activate workflow definitions with SAS Workflow Manager:

1. Develop a draft of your workflow definition. You can create a new workflow definition or import an existing definition and modify it.

2. Save the working draft of your definition. SAS Workflow Manager saves the draft as the Current version. Only you can view and edit your Current version.

3. Create a numbered version of your workflow. Any user that has the appropriate permissions can view numbered versions of workflow definitions.

4. Activate a specific version of the workflow. In order for end-user client applications to be able to create running instances of the workflow, you must activate a specific version of a workflow definition.
Both the Quick Start Tutorial and “Create and Activate a Workflow Definition” on page 8 describe these steps in detail.

Note: Workflow definitions are stored in the BPMN 2.0 XML format.

Figure 1.1 The Workflow Lifecycle

If you determine that you need to make changes to a workflow that has already been activated, you can edit the workflow definition, create a new version, and activate the new version. Client applications can create instances of the new version without affecting any running instances of the old version. Instances of the old version terminate as the workflows are completed.

Only one version of a workflow can be active at a time. As soon as you activate the new version, all new instances of the workflow use the new version of the definition. The older version of the definition is disabled.

Note: Workflow instances are started by the solution. Workflow tasks are managed by the solution while certain administrative activities are available from the Instances tab from within SAS Workflow Manager.

Sign In to SAS Workflow Manager

Note: If you are already signed in to SAS Drive, you can access SAS Workflow Manager by clicking Manage Workflows.

To sign in to SAS Workflow Manager:
1 In the address bar of your web browser, enter the URL for SAS Workflow Manager and press Enter. The Sign In page appears.

Note: Contact your system administrator if you need the URL for SAS Workflow Manager. The default URL is http://host_name/SASWorkflowManager.

2 Enter a user ID and password.

3 Click Sign In.
# Creating and Editing Definitions

*About Workflow Definitions* ........................................... 8

*Create and Activate a Workflow Definition* .............................. 8

*Edit Definition Properties* ........................................ 10

*Edit Element Properties* ........................................ 11

*Specify a Client Identifier* ........................................ 11

*Add Sequence Flows* ................................................... 12

*Add Sequence Flow Conditions* ..................................... 13

*Add Annotations and Associations* .................................. 14

*Manage Permissions for a Definition* ................................ 15

  *Ways to Set Permissions* ........................................ 15

  *Set Permissions on a Workflow Definition* ...................... 15

*Import a Definition* .................................................. 16

*Export a Definition* .................................................. 16

*Create and Manage Data Objects* .................................... 17

  *About Data Objects and Scope* ................................ 17

  *Create a Data Object* ........................................... 17

  *Edit a Data Object* ............................................. 19

  *Delete a Data Object* .......................................... 19

*Creating and Managing Tags* ........................................ 20

  *About Tags* ..................................................... 20

  *Create a Tag* .................................................. 20

  *Add a Tag to a Definition* .................................... 20

  *Delete a Tag* .................................................. 21

*Manage Versions of Workflow Definitions* ......................... 21

  *About Versions* ................................................ 21

  *Set the Displayed Version* .................................... 21

  *Create a Numbered Version* .................................... 22

*Activate a Version* .................................................. 22

*Delete Definitions* .................................................. 23
About Workflow Definitions

A workflow is a series of tasks, together with the participants and the logic that are required to complete the tasks. A workflow definition is a model of a workflow. A workflow definition can include events, gateways, activities, and other elements. SAS Workflow Manager enables you to create a visual model of the workflow definition as a node-link diagram. When you save the workflow definition, it is saved as a Business Process Model Notation (BPMN) file that conforms to the BPMN 2.0 standard.

Create and Activate a Workflow Definition

TIP Before you model your workflow in SAS Workflow Manager, review the workflow patterns described in Chapter 10, "Understanding Workflow Patterns," on page 71. Using the right workflow pattern for the right purpose helps ensure that your workflow is efficient and produces the intended results.

1 Click on the navigation bar.

2 Click New Definition. The New Definition window appears.

3 Enter a name for the definition if you do not want to use the default name. Definition names must start with a letter and can contain only alphanumeric characters and the underscore (_). Definition names are limited to 255 characters and must be unique.

4 (Optional) Enter a description for the new definition.

5 Click OK. SAS Workflow Manager opens the new definition and displays the Definition tab. You define the main process for your workflow on this tab. Any subprocesses that you add to the workflow are displayed on separate tabs.

6 To add an element, drag the element from the list of elements onto the workflow diagram. The Start and End elements are automatically added to the workflow diagram.

   Note: A workflow definition must contain a Start element and at least one End or Terminate End element.

7 Define the data objects that are required by the workflow. You define data objects in the Definition Properties window.

8 (Optional) Configure the Start element.
You configure workflow elements by editing the element properties. See “Edit Element Properties” on page 11.

9 Add and configure the elements required in the workflow. For additional information, see the following topics:

- Chapter 3, “Working with User Tasks,” on page 25
- Chapter 4, “Working with Service Tasks,” on page 35
- Chapter 5, “Working with Subprocesses and Call Activities,” on page 51
- Chapter 6, “Using Timers,” on page 55
- Chapter 7, “Using Messages,” on page 59
- Chapter 8, “Using Gateways,” on page 63
- “Add Annotations and Associations” on page 14

10 Add a Terminate End element to the workflow definition, or keep the End element that was added to the workflow definition during creation.

**TIP** The End element ends the process defined by the workflow path to which it is added, but it allows any parallel paths and subprocesses to complete. The Terminate End element terminates the process and all of its subprocesses.

11 Add sequence flows to connect the elements in the workflow. To connect two elements together, click \( \text{ } \) to enable link mode, and then drag the cursor from the first element to the next.

**TIP** To move elements in the diagram, click \( \text{ } \) to enable move mode. You can switch between link mode (\( \text{ } \)) and move mode (\( \text{ } \)) as needed.

For more information, see “Add Sequence Flows” on page 12.

12 (Optional) Specify conditions for the sequence flows. The conditions on sequence flows are evaluated when a workflow instance executes, and the results determine which paths are taken. See “Add Sequence Flow Conditions” on page 13 for more information.

13 (Optional) Add tags to the definition. Tags enable you to categorize related definitions on a user-defined basis. See “Creating and Managing Tags” on page 20 for more information.

14 Click \( \text{ } \) to save the workflow definition. The definition is saved as the Current version.

**IMPORTANT** An application cannot create instances of the Current version of a definition. In order for an application to use a workflow definition, you must specify a client identifier in the definition properties, create a numbered version, and activate the version.
Specify a client identifier for the definition. The client identifier is the name of the SAS solution that will create instances of the workflow. See “Specify a Client Identifier” on page 11 for more information.

Create a numbered version of the definition. For information, see “Manage Versions of Workflow Definitions” on page 21.

(Optional) Set permissions on the definition. For information, see “Set Permissions on a Workflow Definition” on page 15.

Activate the version. After a numbered version of a definition is activated, the solution that is specified by the client identifier can create new instances of the workflow definition. See “Activate a Version” on page 22 for more information.

IMPORTANT Changes to workflow definitions do not take effect until you activate a revised version of the definition. Instances of a workflow that are already running continue to run with the version of the definition with which they were started. When new workflow instances are started, they use the newly activated version of the definition.

Click Close to close the definition.

Edit Definition Properties

Note: If you have not created a numbered version of a workflow definition, you can edit all of a definition’s properties. After you create a numbered version of a definition, you can edit only the data objects.

Click in the object toolbar, which is located in the top upper right of the window.

The Definition Properties window appears.
2 Edit the definition properties as needed. See “Specify a Client Identifier” on page 11 and “Create and Manage Data Objects” on page 17 for more information.

3 Click OK.

---

**Edit Element Properties**

Click ➔ to hide the properties pane for the selected element.

All elements, except annotation, have a name and description. These properties are displayed on the **Properties** pane. Click ➔ to display the general properties for the selected task.

Element names are limited to 255 characters and descriptions are limited to 1000 characters.

---

**Specify a Client Identifier**

The client identifier is the name of the application that will create instances of the workflow.

1 Click the tab for the main process in the workflow. You cannot specify a client identifier if the current tab is a subprocess tab.

2 Click ➔ in the object toolbar, which is located in the top upper right of the window.

**TIP** If you have not created a numbered version of a workflow definition, you can edit all of a definition’s properties. After you create a numbered version of a definition, you can edit only the description and data objects.

3 Select the client identifier and click OK.
Add Sequence Flows

A sequence flow connects two events, gateways, or activities in a workflow definition. Sequence flows show the order in which the elements in a workflow are executed. A sequence flow is depicted as a solid arrow. For example, each task in the definition shown in Figure 2.1 has one incoming sequence flow and one outgoing sequence flow.

Figure 2.1 Sequence Flows

To connect two elements together, click to enable link mode, and drag the cursor from the first element to the next.

TIP Click or right-click on the element to add the next element by using a pop-up menu.
Add Sequence Flow Conditions

You can specify conditions on most outgoing sequence flows. A condition has the following form:

\[
data \text{ object} \text{ operator} \text{ value-or-data-object}
\]

For example:

\[
\text{Approved} == 'True'
\]

When a workflow instance is run, the condition is evaluated depending on the workflow pattern that the workflow uses. If the condition evaluates to True, the outgoing sequence flow is taken. For more information, see Chapter 10, “Understanding Workflow Patterns,” on page 71.

To add a condition to a sequence flow:

1. Select the sequence flow.
2. (Optional) Enter a name and description for the sequence flow in the properties pane.
   
   **TIP** If you enter a name, it is displayed in the diagram. Names enhance the readability of the diagram and make it easier for others to follow the logic of the workflow.

3. Click \(\ddagger\) to view the **Condition** pane.

4. Enter the condition in the **Condition** field. You can also use the **Data Objects** and **Operators** fields to add object names and operators to the condition.
   
   - Select a data object, and click \(\ddagger\) to add the data object to the **Condition** field.
   - Select the operator, and click \(\ddagger\) to add the operator to the **Condition** field.

For example, the sequence flows from the user task to the service tasks in Figure 2.2 are assigned the conditions shown in **Table 2.1**.
### Table 2.1 Sequence Flow Conditions

<table>
<thead>
<tr>
<th>Sequence Flow Name</th>
<th>Condition</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved</td>
<td>Approved=='True'</td>
<td>Publish model</td>
</tr>
<tr>
<td>Rejected</td>
<td>Approved=='False'</td>
<td>Send email</td>
</tr>
</tbody>
</table>

### Add Annnotations and Associations

An annotation is a note that you can associate with events, activities, and gateways. Annotation elements are connected by an association to the executable element that they describe. An association is depicted by a dotted line.

To add an annotation:

1. Drag the **Annotation** element onto the diagram.
2. On the **Properties** pane for the Annotation, enter the text of the note.
3. Click **Link** to enable link mode.
4. Drag the cursor from the annotation to the element to which it refers.
Manage Permissions for a Definition

Before an authenticated user can work with workflow definitions and instances in SAS Workflow Manager, they must be given the appropriate permissions.

Any authenticated user can create workflow definitions. Users have full permissions to any definition that they create. Members of the Application Administrators group have full permissions to both workflow definitions and workflow instances. The SAS Workflow Editor, SAS Workflow Viewer, SAS Workflow Definition Administrator, and SAS Workflow Process Administrator groups are no longer defined for you. See “Granting Permission to Work with Instances” in SAS Workflow Manager: Administrator’s Guide for more information.

Ways to Set Permissions

Permission to work with workflow definitions and instances is controlled in two ways:

- through group membership. For more information, see “Granting Permission to Work with Instances” in SAS Workflow Manager: Administrator’s Guide.
- by setting permissions on specific workflow definitions. For more information, see “Set Permissions on a Workflow Definition” on page 15.

Set Permissions on a Workflow Definition

| Note: Permissions that you set on specific workflow definitions override permissions that you set through group membership. |

To set permissions on a workflow definition in SAS Workflow Manager:

1 In the Definitions view, open the definition for which you want to set permissions.

2 Click  and select Authorization. The Authorization window appears.

3 (Optional) Add an identity.
   a Click . The Select Identity window appears.
   b Select User or Group depending on the type of identity that you want to add.
   c In the search field, enter the identity or enter an asterisk (*) to display the list of identities.
   d Select the identity that you want to add, and click OK.

4 (Optional) Delete an identity. To delete an identity, select the identity and click .
5 Modify the permissions for the identities. Select or clear the check boxes to set or remove the appropriate permissions.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Abilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>View the definition.</td>
</tr>
<tr>
<td>Update</td>
<td>Modify and delete the definition. Create new versions of the definition.</td>
</tr>
<tr>
<td>Enable</td>
<td>Modify, delete, and activate the definition. Create new versions of the definition.</td>
</tr>
<tr>
<td>Secure</td>
<td>Modify, delete, and activate the definition. Create new versions of the definition, and set permissions on the definition.</td>
</tr>
</tbody>
</table>

6 Click **Save** to save the permissions and close the Authorization window.

---

### Import a Definition

1 On the Definitions category page, click **Import**. The Import Definition window appears.

2 Click **Import**.

3 Select the Business Process Model and Notation (BPMN) file that you want to import, and click **OK**.

---

### Export a Definition

You can export workflow definitions as Business Process Model and Notation (BPMN) files.

1 Open the definition that you want to export.

2 Click **Export** and select **Export**. SAS Workflow Manager adds the BPMN file to the download bar in your browser.
Create and Manage Data Objects

About Data Objects and Scope

Data objects are similar to variables. They hold pieces of business data that are required by the workflow. Data objects that are defined in the main process of a definition are global data objects and can be used by elements anywhere in the workflow. Data objects that are defined in a subprocess are local data objects and can be used only within the subprocess and its children.

**Note:** Do not duplicate application data structures in data objects. Instead, define only the data that is referenced in the workflow definition elements. This practice results in a more efficient workflow that is less likely to be affected by changes in the application data model.

Create a Data Object

You can define new data objects by using the Definition Properties window, or you can select the **Create a new data object** option in most fields where you can specify a data object.

To create a data object by using the Definition Properties window:

1. Ensure that the correct tab is displayed. To define global objects, click the tab for the main process. To define local data objects, click the tab for the subprocess.
2. Click in the object toolbar, which is located in the top upper right of the window. The Definition Properties window appears.
3. Click above the data objects table. The New Data Object window appears.
4. Enter a name for the data object. Data object names must start with a letter and can contain only alphanumeric characters and the underscore (_). Data object names are limited to 50 characters and must be unique within a workflow.
5. Select the data type for the data object.
6. (Optional) Specify a default value for the data object.
7. Click **Save**.
8. Click **OK** to close the Definition Properties window.

To specify or create a new data object by using the **Create a new data object** option in a field in which a data object table exists:

1. Click , and then select **Data object** in the window that appears.
In the drop-down menu, select **Create a new data object**. The New Data Object window appears.

Enter a name for the data object. Data object names must start with a letter and can contain only alphanumeric characters and the underscore (_). Data object names are limited to 50 characters and must be unique within a workflow.

Select the data type for the data object.

(Optional) Specify a default value for the data object.

**IMPORTANT** If a datetime type data object does not have a default value, it is set to the current datetime since the value cannot be null. You must initialize datetime data objects before they are referenced in any workflow processes.

Click **Save** to close the New Data Object window.

Click **OK**.
Edit a Data Object

1. Ensure that the correct tab is displayed. To edit a global data object, click the tab for the main process. To edit a local data object, click the tab for the subprocess in which the data object is defined.

   **TIP** You can also edit data objects in any field in which a data object table exists by clicking 📊.

2. Click 📊 in the application bar above the diagram. The Definition Properties window appears.
3. Select the data object, and click 🕒. The Edit Data Object window appears.
4. Edit the data object properties as needed.
5. Click **Save**.
6. Click **OK** to close the Definition Properties window.

Delete a Data Object

**Note:** You cannot delete a data object that is used by an element in the definition.

1. Ensure that the correct tab is displayed. To delete global data objects, click the tab for the main process. To delete local data objects, click the tab for the subprocess.

   **TIP** You can also delete data objects in any field where a data object table exists by selecting the data object in the table and clicking 📊.

2. Click 📊 in the application bar above the diagram. The Definition Properties window appears.
3. Select the data object, and click 🕒.
Creating and Managing Tags

About Tags

Tags are alphanumeric labels that are associated with definitions. Tags enable you to categorize related definitions on a user-defined basis. For example, you can define a tag for workflow definitions that is based on the application or organization that owns the definitions.

- Tags are static labels. They have no effect on active workflow instances and cannot be modified at run time.
- Tags do not control access to a definition. They do not prevent the sharing of workflow definitions among different applications. They do not affect whether or how a workflow is active.
- Tags are defined at service level, and you can assign them to any definition in that environment. If a definition is moved to another environment, you are prompted to create the relevant tags in the new environment when you create a numbered version of the definition.

Create a Tag

2. Click . The New Tag window appears.
3. Enter the name of the new tag, and click Save.
4. Click Close to close the Manage Tags window.

Add a Tag to a Definition

Note: You must have permission to edit a definition in order to add a tag to it. See “Manage Permissions for a Definition” on page 15 for more information.

1. Open the definition to which you want to add tags.
2. Click in the application bar above the diagram, and select Tags. The Select Tags window appears.
3. Select the tags that you want to add to the definition, and click OK.
Delete a Tag

1. On the Definitions category page, click \( \text{Manage tags} \) and select Manage tags. The Manage Tags window appears.

2. Select the tag that you want to delete, and click \( \text{Delete} \).

3. Click Close to close the Manage Tags window.

**TIP**
To remove tags from a definition, navigate to the Select Tags window and deselect the tags that you want to remove from the definition.

---

Manage Versions of Workflow Definitions

About Versions

The most recent version of a definition is the Current version. Only you can see your Current version. Other users cannot see changes that you have made to a definition until you create a numbered version of the definition.

**IMPORTANT**
Changes to workflow definitions do not take effect until you activate a revised version of the definition. Instances of a workflow that are already running continue to run with the version of the definition with which they were started. When new workflow instances are started, they use the newly activated version of the definition.

---

Set the Displayed Version

The displayed version is the version whose information is displayed on the Definition tab. On the Versions tab, a \( \text{✓} \) indicates the displayed version. To
change the displayed version, select the version that you want to view and select **Set Version**.

---

**Create a Numbered Version**

**Note:** In order to create a numbered version of the definition, you must specify a **Client identifier** in the Definition Properties window.

**Note:** You cannot save changes to a numbered version. If you edit a numbered version and click **Set Version**, SAS Workflow Manager your changes are saved as the **Current version**. If a Current version does not exist, SAS Workflow Manager creates it. If a Current version already exists, it is replaced.

To create a new version:

1. Click  to create a static numbered version of the workflow definition. The New Version window appears.

2. (Optional) Enter any notes that you want to associate with the new version.

   **TIP** Notes can help you identify what has changed from one version to the next.

3. Click **Yes**. SAS Workflow Manager assigns the number 1.0 to the version.

---

**Activate a Version**

Only one version of a workflow definition can be activated at a time. If a version is already active and you activate a different version, the previously active version is disabled and cannot be used by other applications.

**IMPORTANT** Changes to workflow definitions do not take effect until you activate a revised version of the definition. Instances of a workflow that are already running continue to run with the version of the definition with which they were started. When new workflow instances are started, they use the newly active version of the definition.
A workflow definition is not fully validated until you save it as a version.

**Note:** You cannot activate the Current version of a definition. To activate a definition, you must first create a numbered version. See “Manage Versions of Workflow Definitions” on page 21 for more information.

To activate a numbered version of a definition, open the version, and click ➡️. The version is marked **Active** in the **Status** column.

---

**Delete Definitions**

**Note:** You cannot delete a definition after it has been activated.

1. Click 🗑️ on the navigation bar.
2. Select the check box for the definitions that you want to delete, click ✗️, and then select **Delete**. A confirmation message is displayed.
3. Click **Yes** to delete the workflow definition, or click **No** to exit the message.
About User Tasks

User tasks are tasks that are performed by a human participant. Typically, the user tasks in a running workflow are managed through a software application that provides the ability to claim and complete the tasks. For example, SAS Model Manager provides a Tasks category through which users can claim and complete the tasks to which they are assigned. When workflow execution reaches a user task, an instance of the task is created and each potential owner who has access to the task must complete it. The user task then appears in each potential owner’s task list.

Note: Some workflows can have a separate list for claimed tasks. If a task has multiple owners, that task appears only in the task list of the owner who claimed the task.
Add a User Task

1. Drag the **User Task** object from the list of elements onto the workflow diagram.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. (Optional) Select **Set a due date** and specify a due date.
   - To enter the due date that is relative to when the task starts, select **When the user task is reached**. If you do not enter a delay, the task becomes due as soon as the task starts. You can select a delay under **Add this delay**.
   - To use a data object to specify the due date, select **Specified in the following data object**, and select the data object. The data object is evaluated when the user task starts and does not change even if the value of the data object changes.

   **TIP**  The due date is a task property that is available for use by the application that starts the workflow. The due date does not affect the execution of the workflow. To set deadlines that affect the execution of the workflow, use a timer. See Chapter 6, “Using Timers,” on page 55 for more information.

4. Add the participants for the task. For more information, see “Managing Participants” on page 26.

5. (Optional) Add any prompts that are needed for the task. For more information, see “Managing Prompts” on page 33.

Managing Participants

About Participants and Roles

When you define a user task, you associate users with the task by adding them to the task as participants. A participant can be an individual user or a group of users.

You can assign a participant to one of two roles:

**Potential Owner**
- A potential owner can claim and complete a task.

**Excluded Owner**
- An excluded owner cannot claim or view a task. Excluded owners are typically used when a task needs to be reviewed or approved by at least two people.
Using excluded owners ensures that the same person does not review the task twice and can increase the transparency of an organization’s processes.

**Note:** SAS Workflow supports an additional role: Actual Owner. A participant becomes an actual owner after they claim a task in a running workflow, but you cannot assign this role in a workflow definition.

Participants must be available through the SAS Viya Identities service and have the appropriate permissions. For additional information, see *SAS Workflow Manager: Administrator’s Guide* and *SAS Viya Administration: Identity Management*.

---

**Add Participants to a User Task**

1. Click **Participants** to display the **Participants** properties pane.

   ![Participants properties pane](image)

2. Specify the participants for the task either by using a data object or by selecting user or group identities. Typically, a user task has a set of potential owners, but only one person can claim and complete the task. You can specify one or more users or groups as potential users.

   **TIP**

   If the set of potential owners is unknown at design time, you should use a data object to configure participants dynamically. See “Specifying Groups as Participants” on page 32 for more information.

**Note:** The data object is evaluated when the user task starts and does not change even if the value of the data object changes.
To specify participants by using a data object, select **Listed in this data object**, and select the data object. See “Data Object Format for Participants” on page 31 for more information.

To specify participants by selecting identities, select **Listed in this table**, and complete the following steps for each participant:

a Select the workflow role for the participant. For more information, see “About Participants and Roles” on page 26.

b Click ✏ next to the chosen role and add an owner user or group. The Choose Members window appears.

c In the search field, enter the **User name or Group name**.

TIP You can enter * (an asterisk) to display the list of all available identities.

d Select **User name or Group name**, and click OK to close the Choose Members window.

TIP If you know the user ID of the user or group that you would like to assign as a potential owner, you can enter it directly into the **Select the members** field.

Note: You can add more than one **User name or Group name** at a time.

3 If only one of the potential owners is needed to complete the task, select the **Create a single instance** option. This is automatically selected in the **Participants** properties pane.

(Optional) You can configure the single instance to set or exclude an owner based on a data object or a transient variable. To configure these settings:

a Select **Set actual owner value** or **Set excluded owner value** as the participant role.

b Click ✉ to open the Actual Owner window.

c Do one of the following:

- Select **Data object**, and choose a data object from the drop-down menu to set the actual owner based on the data object.

- Select **Transient variable** and enter the variable information in the text box.

d Click OK to set these changes.
4. (Optional) If more than one potential owner must complete the task, the workflow service must create a separate instance of the task for each person. **Create a single instance** is automatically selected in the Participants pane to enable more than one potential owner to complete the task. You can configure the single instance to set or exclude an owner based on a data object or a transient variable. To configure these settings:

   a. Select **Create a new instance for each participant**. For more information, see "Enabling Multiple Participants to Complete a Task" on page 30.

   b. Select **Run in the instances in parallel** or **Run the instances sequentially**.

   c. (Optional) Specify a completion condition. This condition is an expression that controls how multiple instances are handled or how the user task is determined to be completed. For more information, see "Enabling Multiple Participants to Complete a Task" on page 30.

   For descriptions of the variables that you can use in the expression, see "Completion Conditions and Instance Variables" on page 32.

   To specify a completion condition:

   i. Select an instance variable, and then click † to add the variable to the Completion condition field.

   ii. Select the operator, and then click ‡ to add the operator to the Completion condition field.
iii For the second operand in the expression, either enter a literal value or add a second instance variable to the Completion condition field.

Note: When the Create a new instance for each participant option is selected, the workflow definition diagram indicates that this option is configured.

Delete or Edit a Participant

On the Participants property pane, click × next to the participants name to delete the participant. For additional information about adding a participant, see Step 2a through Step 2d of "Add Participants to a User Task" on page 27.

Enabling Multiple Participants to Complete a Task

Typically, a user task has multiple potential owners, but only one person is required to claim and complete the task. In this case, only one instance of the task is required. Do not select Create a new instance for each participant when you define the task.

If a user task has multiple potential owners and more than one person must claim and complete the task, select Create a new instance for each participant. A new instance of the task is created for each person that is assigned to it. A task for which multiple instances are created is referred to as a multi-instance task. You can use multi-instance tasks to repeat activities in a workflow without defining the same task multiple times.

For multi-instance tasks, you can control the number of instances that are created or when the task is considered complete by specifying a completion condition. Table 3.1 on page 31 lists the criteria for determining when a multi-instance task is complete.

Note: When you specify a group as a potential owner, the number of potential owners that are assigned to a task and the number of instances that are created depend on how you specify the group. For more information, see “Specifying Groups as Participants” on page 32.
Table 3.1 Criteria for Completing Multi-instance User Tasks

<table>
<thead>
<tr>
<th>Create a new instance for each participant is selected</th>
<th>Completion condition is specified</th>
<th>Criteria for Determining When a Task Is Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>The completion condition evaluates to True, and the required number of potential owners claim and complete the task.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Every potential owner claims and completes the task.</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>One of the potential owners claims and completes the task.</td>
</tr>
</tbody>
</table>

For more information, see “Completion Conditions and Instance Variables” on page 32 and Step 4 of “Add Participants to a User Task” on page 27.

Data Object Format for Participants

To specify participants by using a data object, the value of the data object must follow this format:

```json
[{
  "principalType" : "type",
  "role" : "role",
  "principal" : "identity"
}
<,{
  "principalType" : "type",
  "role" : "role",
  "principal" : "identity" }
...>]
```

**type**

specify user for individual users and group for groups.

**TIP** If more than one member of a group needs to complete a multi-instance task, the data object must list the name of the group once for each member that must complete the task. See “Specifying Groups as Participants” on page 32 for more information.

**role**

specify PotentialOwner or ExcludedOwner. See “About Participants and Roles” on page 26 for descriptions of these roles.

**identity**

specify an identity that is available through the SAS Viya Identities service. See SAS Viya Administration: Identity Management for additional information.
Specifying Groups as Participants

If you specify a group name only once, then only one member of the group can claim and complete the task. The first member to claim the task becomes the actual owner and can complete the task. The remaining members of the group can no longer claim the task.

If more than one member of a group needs to complete a task, then you must specify the potential owners by using a data object with a multi-instance task. The data object must list the name of the group once for each member that must complete the task. For example, if the group Approvers defines ten users, and three users must claim and complete the task, then the data object must contain the following value:

```json
[{ "principalType" : "Group", "role" : "PotentialOwner", "principal" : "Approvers" },
 { "principalType" : "Group", "role" : "PotentialOwner", "principal" : "Approvers" },
 { "principalType" : "Group", "role" : "PotentialOwner", "principal" : "Approvers" }]
```

Completion Conditions and Instance Variables

For multi-instance user tasks, you can specify a condition that indicates when the task is complete. SAS Workflow Manager provides three predefined variables that you can use to construct a completion condition:

- **numberOfInstances**
  - specifies the total number of active and completed instances for the task. For example, if you specify a completion condition of `numberOfCompletedInstances>=numberOfInstances/2`, the task is considered complete when half of the potential owners have claimed and completed the task.

- **numberOfActiveInstances**
  - specifies the total number of instances that can be active at the same time. If the instances are run sequentially, this number must be equal to 1. If the instances are run in parallel, this number cannot exceed the value of `numberOfInstances` (if it is specified).

  For example, suppose you have a user task that has ten potential owners and the instances for each user are run in parallel. You can limit the number of people that can claim the task at the same time to two by specifying the following condition:

  `numberOfActiveInstances==2`

  As soon as one of the two participants completes the task, a third participant can claim and complete the next instance of the task.

- **numberOfCompletedInstances**
  - specifies the number of instances that must be completed. This number cannot exceed the value of `numberOfInstances` (if it is specified).

  For example, suppose you have a user task that has five potential owners, but only three people need to complete the task. You would specify the following condition:
Managing Prompts

About Prompts, Scope, and Reuse

Prompts enable you to request information from workflow participants and store the user’s response in data objects. By using prompts, you can set values that are required to evaluate sequence flow expressions or to execute other tasks.

You can define prompts for start elements and user tasks. You must define prompts for required values before the values are needed. For example, if a data object value needs to be initialized when a workflow instance is created, define a prompt on the start element. If a value is required by the sequence flow expressions for a gateway, define the prompt on a start element or on a task that is earlier in the workflow than the gateway.

A prompt has the same scope as the data object that is used to store the user’s response. Depending on the scope of the data object, a prompt that is defined for one element can be reused by (added to) other user tasks. If the scope of the data object is global, the prompt can be reused anywhere within the workflow definition. If the scope is local, the prompt can be reused only within the process where it is defined and within any children of that process. See “About Data Objects and Scope” on page 17 for more information.

Define a New Prompt

1. Select the Start element or User task for which you want to define a prompt.
2. Click ☐ to display the Prompts property pane.
3. Click ☐. The New Prompt window appears.
4. In the Prompt label field, enter the text that you want to be displayed for the prompt.
5. (Optional) Select This value is required if the user is required to enter or select a value for this prompt.
6. Select or create the data object in which you want to store the user’s response. See “Create a Data Object” on page 17 for additional information.
7. (Optional) Specify the list of valid display values for the response. To add a new display value:
   a. Click ☐. SAS Workflow Manager adds a new row to the Displayed values table.
In the **Displayed values** table, enter a name and value.

**TIP** To delete a prompt value, select the value in the table and click —.

1. Select the **Start** element or **User task** for which you want to define a prompt.
2. Click to display the **Prompts** property pane.
3. Click above the prompt table. The Add Prompts window appears.
4. Select the prompts that you want to add, and click **OK**.

---

**Reuse an Existing Prompt**

---

**Delete or Edit a Prompt**

On the **Prompts** property pane, select the prompt in the tablet, and click either ⬅️ or ↗️. For additional information about editing a prompt, see Step 3 through Step 8 of “Reuse an Existing Prompt” on page 34.

---

**Note:** You can add prompt values to or remove them from an existing prompt definition without affecting previous associations.
About Service Tasks

A service task is a task that invokes an executable business action. These actions are triggered automatically when workflow execution reaches the service task. For example, you can use service tasks to perform the following actions:

- send email to process participants and groups about deadlines or status changes
- communicate with other applications by executing web services
- execute business logic or rules
- integrate the system with other back-end systems
When you define a service task, you select one of seven different actions:

**Invoke REST web service**
- invokes a REST web service

**Invoke web service**
- invokes a SOAP web service

**Send email**
- sends email using the SAS Viya Mail Service

**Send notification**
- enables you to send separate notifications to individual users or groups using the SAS Viya Notification Service

**Send group notification**
- enables you to send a single notification to individual users and groups, and to copy additional users and groups using the SAS Viya Notification Service

**Invoke job execution**
- enables you to execute jobs that are defined in the Job Execution service

**Select a template**
- enables you to select an existing template.

---

**About Service Task Transient Variables**

Transient variables can be used as process variables to pass data. Use transient variables only to store values that are required for the next step in the workflow. For example, you can use a transient variable when a web service response contains a value that is required as input to another web service call.

Transient variables are identified by an underscore prefix.

**Note:** Only use transient variables to pass data from within a workflow that is not part of the solution state.

---

**Add a Service Task**

1. Drag the Service Task element from the list of elements onto the workflow diagram.
2. (Optional) On the Properties pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.
3. Click 📚 to display the Action property pane.
4. Select the action.
5. Enter the information for the service task. See the following topics:
The Invoke REST Web Service Action

This action enables you to invoke a REST web service over HTTP.

Table 4.1 Properties for the Invoke REST Web Service Action

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Specifies the URL of the web service endpoint for the task.</td>
</tr>
<tr>
<td>Method</td>
<td>Specifies the HTTP method for the web service such as GET, POST, PUT, and so on.</td>
</tr>
<tr>
<td>HTTP request headers</td>
<td>Specifies the names and values of the HTTP header fields in the request. Click + to add a new header. To remove or edit a header, select the header and click either - or .</td>
</tr>
<tr>
<td>HTTP response header</td>
<td>Specifies the data object mapping to field values that are part of the HTTP header in the response. Click + to add a new header. To remove or edit a header, select the header and click either - or .</td>
</tr>
<tr>
<td>Expected result code</td>
<td>Specifies the HTTP status code that indicates that the request was successful.</td>
</tr>
<tr>
<td>Input type</td>
<td>Specifies the type of the input body. The type can be JSON, XML, or empty (no input body).</td>
</tr>
<tr>
<td>Input body</td>
<td>Specifies the content of the input body of the request.</td>
</tr>
<tr>
<td>Output type</td>
<td>Specifies the output type of the response. The type can be either JSON or XML.</td>
</tr>
</tbody>
</table>
Output data objects

Associates data objects with result fields in the response. See “Manage Output Data Objects for REST Web Service Tasks” on page 38 for instructions.

**IMPORTANT** A user account must be configured for a workflow client in order to make service task calls within a definition. This account must be an LDAP account and must be configured by the solution at start-up, or it must be configured by the system administrator prior to the creation of any instances. For more information, see “Configuring the Workflow Client” in SAS Workflow Manager: Administrator’s Guide.

**Note:** External requests are not supported at this time. Only relative URLs are supported for the URL field. To prevent unauthorized platforms actions, the following URIs are restricted:

- /workflow
- /SASLogon
- /credentials
- /identities

---

**Manage Output Data Objects for REST Web Service Tasks**

To associate a data object with a result field in the response:

1. Click \(\text{Add}\). The Add an Output Data Object window appears.
2. Select the name of the data object, or select **Create a new data object**. See “Create a Data Object” on page 17 for more information.
3. Enter the name of the result field in the HTTP response that contains the information that you want to store in the data object. If the output type is JSON, specify the name using JSON path expression syntax. If the output type is XML, specify the name using an XPath expression.
4. Enter the path expression that is required to extract the value of the output parameter from the response.
5. Select the data type of the value of the target output parameter.
6. Click **OK**.

To remove or edit a data object, select the data object, and click either \(\text{Delete}\) or \(\text{Edit}\).
Invoke REST Web Service Action Example

This example illustrates a basic usage scenario for an Invoke REST Web Service Task.

Table 4.2  Invoke REST Web Service Parameter Summary

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td><code>/workflow-sample/rest/creditCheck?id=${id}</code></td>
</tr>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>HTTP request readers</td>
<td>Enter <code>Content-Type</code> for the <strong>Key</strong> and <code>application/json</code> for the <strong>Value</strong>.</td>
</tr>
<tr>
<td></td>
<td>Enter <code>Accept</code> for the <strong>Key</strong> and <code>application/json</code> for the <strong>Value</strong>.</td>
</tr>
<tr>
<td>Expected result code</td>
<td>200</td>
</tr>
<tr>
<td>Input type</td>
<td>Empty</td>
</tr>
<tr>
<td>HTTP response headers</td>
<td><code>outputHeader</code></td>
</tr>
<tr>
<td>Output data objects</td>
<td>Select <code>rating</code> for the <strong>Data object</strong> and enter <code>$rating</code> for the path expression.</td>
</tr>
<tr>
<td></td>
<td>Select <code>name</code> for the <strong>Data object</strong> and enter <code>$name</code> for the path expression.</td>
</tr>
</tbody>
</table>

The following table provides the expected values of the workflow instance data objects after the credit check REST service execution.

Table 4.3  Invoke REST Web Service Results Summary

<table>
<thead>
<tr>
<th>Data Object</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rating</td>
<td>575</td>
</tr>
<tr>
<td>name</td>
<td>John Doe</td>
</tr>
<tr>
<td>outputHeader</td>
<td>“123”</td>
</tr>
</tbody>
</table>
The Invoke Web Service Action

This action enables you to invoke a SOAP web service.

Table 4.4 Properties for the Invoke a Web Service Action

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Specifies the URL of the web service endpoint.</td>
</tr>
<tr>
<td>Action</td>
<td>Specifies the URI of the SOAPAction header field.</td>
</tr>
<tr>
<td>XML input document</td>
<td>Specifies the content of the XML input body.</td>
</tr>
<tr>
<td>Output data objects</td>
<td>Associates data objects with XPath expressions that extract information from the SOAP response. See “Manage Output Data Objects for SOAP Web Service Tasks” on page 40 for instructions.</td>
</tr>
</tbody>
</table>

Manage Output Data Objects for SOAP Web Service Tasks

To associate a data object with a path expression in the responses:

1. Click 📊. The Add an Output Data Object window appears.
2. Select the name of the data object, or select Create a new data object. See “Create a Data Object” on page 17 for more information.
3. Enter the path expression that is required to extract the value of the output parameter from the SOAP response.
4. Select the data type of the value of the target output parameter.
5. Click OK.

To remove or edit a data object, select the data object, and click either ⌰ or ⚙️.
Invoke Web Service Action Example

This example provides a basic usage scenario for an Invoke Web Service Task.

**Table 4.5  Invoke Web Service Parameter Summary**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td><a href="http://localhost/workflow-sample/creditCheck">http://localhost/workflow-sample/creditCheck</a></td>
</tr>
<tr>
<td>Action</td>
<td><a href="http://localhost/workflow-sample/service/getCreditRequest">http://localhost/workflow-sample/service/getCreditRequest</a></td>
</tr>
<tr>
<td>XML input document</td>
<td>&lt;ns2:getCreditRequest xmlns:ns2=&quot;<a href="http://workflow-sample/service%22%3E$">http://workflow-sample/service&quot;&gt;$</a> {id}&lt;/ns2:id&gt;&lt;/ns2:ud&gt;&lt;/ns2:getCreditRequest&gt;</td>
</tr>
</tbody>
</table>

**Output data objects**

Select **rating** for the **Data object** and enter /ns2:getCreditResponse/ns2:creditReport/ns2:creditRating/text() for the path expression.

Select **name** for the **Data object** and enter /ns2:getCreditResponse/ns2:creditReport/ns2:name/text() for the path expression.

The following table documents the expected values of the workflow instance data objects after the credit check REST service execution.

**Table 4.6  Invoke SOAP Web Service Results Summary**

<table>
<thead>
<tr>
<th>Data Object</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rating</td>
<td>525</td>
</tr>
<tr>
<td>name</td>
<td>John Doe</td>
</tr>
<tr>
<td>outputHeader</td>
<td>123</td>
</tr>
</tbody>
</table>
The Send Email Action

This action enables you to send email using the SAS Viya Mail Service. For more information, see “Mail Service” in SAS Viya Administration: Configuration Properties.

Table 4.7  Send an Email Action Properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>Specifies the recipients for the mail message. You can list each user or group individually in the service task properties. Select <strong>Listed in this field</strong> and click <img src="image" alt="List" /> to select the users or groups. You can specify a data object that contains the list of recipient IDs. Select <strong>Listed in this data object</strong> and select the data object from the drop-down list. <strong>Note:</strong> If you want to refer to a data object’s value within an email notification, you must use the <code>${data_obj_name}</code> JUEL syntax.</td>
</tr>
<tr>
<td>Subject</td>
<td>Specifies the subject of the email message.</td>
</tr>
<tr>
<td>Message</td>
<td>Specifies the message text.</td>
</tr>
</tbody>
</table>

**Note:** The sender’s email address is populated by the email service with the system account information. You cannot modify the sender’s email address in SAS Workflow Manager.

The Send Email Action Example

This example provides a basic usage scenario for a Send Email Task.

Table 4.8  Send Email Parameter Summary

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>user IDs</td>
</tr>
<tr>
<td>Subject</td>
<td>Email Test</td>
</tr>
<tr>
<td>Message</td>
<td>This is a test of the mail service.</td>
</tr>
</tbody>
</table>
The Send Notification Action

This action enables you to send separate notifications to individual users or groups using the SAS Viya Notification Service.

Note: The notification channel is determined by the user preference settings. Supported channels include store, email, and SMS messages. The portal retrieves stored messages.

Table 4.9  Send Notifications Properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>To</strong></td>
<td>Specifies the recipients for the message. You can list each recipient or group individually in the service task properties, specify a data object that contains the list of recipients, or notify only task owners. When you select <strong>Recipients or group recipients</strong>, you add users or groups by clicking 🔄. When you select <strong>Listed in this object</strong>, select <strong>User</strong> or <strong>Group</strong> and select the data object from the drop-down list. Note: If the task is unclaimed, the message goes to all potential owners. If the task is claimed, the message goes only to the owner who claimed the task.</td>
</tr>
<tr>
<td><strong>Template</strong></td>
<td>Specifies the notification template that you want to use for the notification. The notification template is managed by the Template service.</td>
</tr>
<tr>
<td><strong>Template Mapping Variables</strong></td>
<td>Specifies the template variable to data object mapping.</td>
</tr>
</tbody>
</table>

The Send Notification Action Example

This example provides a basic usage scenario for a Send Notification Task.
The Send Group Notification Action

This action enables you to send a notification to individual users and groups where all recipients are addressed together in a single message. You can also copy additional users and groups.

**Table 4.11   Send Group Notifications Properties**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>To</strong></td>
<td>Specifies the recipients for the mail message. You can list each recipient or group individually in the service task properties, specify a data object that contains the list of recipients, or notify only task owners. When you select <strong>Recipients or group recipients</strong>, you add users or groups by clicking 📩. You can also add users or group to the <strong>CC</strong> and <strong>BCC</strong> fields. When you select <strong>Listed in this object</strong>, select <strong>User</strong> or <strong>Group</strong> and select the data object from the drop-down list. You can also add data objects to the <strong>CC users</strong>, <strong>CC group</strong>, <strong>BCC users</strong>, and <strong>BCC group</strong> fields. <strong>Note:</strong> If the task is unclaimed, the message goes to all potential owners. If the task is claimed, the message goes only to the owner who claimed the task.</td>
</tr>
<tr>
<td><strong>CC</strong></td>
<td>Specifies one or more recipients. You can list each recipient or group individually in the service task properties or specify a data object that contains the list of recipients.</td>
</tr>
</tbody>
</table>
The Send Group Notification Action Example

This example provides a basic usage scenario for a Send Group Notification Task.

Table 4.12  Send Workflow Group Notification Parameter Summary

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>user IDs</td>
</tr>
<tr>
<td>CC</td>
<td>Application Administrators</td>
</tr>
<tr>
<td>BCC</td>
<td>SAS Administrators</td>
</tr>
<tr>
<td>Template</td>
<td>SAS_Email_HTML_Message</td>
</tr>
<tr>
<td>Template mapping variables</td>
<td></td>
</tr>
</tbody>
</table>

Select DESC for the Template Variable and enter Testing 123 for the Data Object field text.

Select TO for the Template Variable and enter dear sir or madam for the Data Object field text.

Select MSG for the Template Variable. For the Data Object field, enter This is a test of the group notification service.

Note: Email is the only delivery channel supported by using the group notification policy.
The Invoke Job Execution Action

This action enables you to execute code that has been registered with the SAS Viya Job Execution Service. A single job can execute a single task or multiple tasks bundled together.

Table 4.13  Invoke Job Execution Properties

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Definition</td>
<td>Specifies the available job definitions. Select the Job Definition from the drop-down list.</td>
</tr>
<tr>
<td>Arguments</td>
<td>Specifies the arguments defined for the selected job definition. The user must specify a value for each job definition if no default value is provided. When the job definition has been selected, you can view the defined arguments. Select the argument and click to specify the arguments value.</td>
</tr>
<tr>
<td>Results</td>
<td>Specifies the output values and associated data objects. Click to open the Add a Result Data Object window. Select Data object and select the data object from drop-down list. You can also select and enter a transient variable. You must specify a Result name. Click OK to close the window. To remove or edit a result, select the result and click either or .</td>
</tr>
<tr>
<td>Result State</td>
<td>Specifies the data object used to store the job execution status. Click to open the Result State window. Select Data object and select the data object from the drop-down list. You can also select and enter a transient variable. Click OK to close the window.</td>
</tr>
</tbody>
</table>

Note: In order for the job definition to be displayed in the Job Definition list, you must define the job definition with the appropriate access for all registered SAS applications or with SAS Workflow Manager. Job definitions created using the SAS Job Execution Web Application should have the required authorization. For more information, see “Overview” in SAS Job Execution Web Application: User’s Guide.

IMPORTANT  The job definition must contain a parameter named _contextName. Assign a valid compute context to this parameter at design time.
A user account must be configured for a workflow client in order to make service task calls within a definition. This account must be an LDAP account and also a host user account for invoking job executions. This account should be configured by the solution at start-up, or by the system administrator prior to the creation of any instances. For more information, see “Configuring the Workflow Client” in SAS Workflow Manager: Administrator’s Guide.

The Invoke Job Execution Action Example

This example provides a basic usage scenario for an Invoke Job Execution Task.

Table 4.14  Submit JES Job Parameter Summary

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Definition</td>
<td>JobTest</td>
</tr>
</tbody>
</table>

Arguments

Select the \texttt{var} data object for the \texttt{Name}, and select the \texttt{$category} [1] data object for the \texttt{Value}.

Select the \texttt{_contextName} data object for the \texttt{Name}, and select the SAS Job Execution compute context data object for the \texttt{Value}.

Results

Select the \texttt{total} data object for the \texttt{Name}, and select the \texttt{result} [2] data object for the \texttt{Value}.

Result State

Select \texttt{jobStatus} [3] for the Data object.

The Template Action

This action enables you to specify a full or partial Invoke REST Service Task definition and save it as a custom task template. Once a task template has been created, you can select it for use in other workflow definitions.

Create a Template

1. Follow the steps in “Add a Service Task” on page 36 to create an Invoke REST Service task.
2 Click ✉. The **Save As Template** window appears.

3 Specify a name for the task template.

4 (Optional) Enter a description for the task template.

5 Select the associated application in the drop-down menu.

   **Note:** The associated application that you select should be the same as the Client Identifier for the workflow definition.

6 (Optional) Specify the data objects that you want to save in the template. For more information, see “Create and Manage Data Objects” on page 17.

   **Note:** You cannot create new data objects to associate with the template. Data objects must already exist in the workflow definition.

7 Click **OK** to save the template.

**Use a Template In a Workflow Definition**

1 Drag the **Service Task** element from the list of elements onto the workflow diagram.

2 (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3 Click ✉ to display the **Action** property pane.

4 Select **Select a template** in the **Action type** drop-down menu. The Select a Template window appears.

5 Select the template that you want to use, and click **OK**.

   All of the fields in the **Actions** pane are populated with the information stored in the template. A confirmation message appears, and the workflow template is now in use.
6 Click [ ] to save your workflow definition.
About Subprocesses and Call Activities

Subprocesses and call activities are ways to modularize and reuse workflow logic. Subprocesses and call activities both define workflow processes. Subprocess definitions are embedded in the main workflow definition. Call activities are defined separately and referenced from the main workflow definition. For a running workflow instance, a subprocess runs within the same process as the main process. Call activities run as separate processes.

Using subprocesses and call activities improves readability of a workflow definition and promotes consistent reuse of business logic within an organization.

Add a Subprocess

1. Drag the **Subprocess** from the list of objects onto the workflow diagram.
2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.
3. Click [+] on the subprocess to open it on a separate tab.
4. To add an element, drag the element from the list of elements onto the subprocess diagram. The **Start** and **End** nodes are automatically added to the element.

**Note:** A subprocess definition must contain a **Start** element and at least one **End** or **Terminate End** element.
5 Define the data objects that are required by the subprocess. You define data objects in the Definition Properties window.

6 (Optional) Configure the Start element.
   You configure subprocess elements by editing the element properties. See “Edit Element Properties” on page 11.

7 Add and configure the elements required in the subprocess. For additional information, see the following topics:
   - Chapter 3, “Working with User Tasks,” on page 25
   - Chapter 4, “Working with Service Tasks,” on page 35
   - Chapter 5, “Working with Subprocesses and Call Activities,” on page 51
   - Chapter 6, “Using Timers,” on page 55
   - Chapter 7, “Using Messages,” on page 59
   - Chapter 8, “Using Gateways,” on page 63
   - “Add Annotations and Associations” on page 14

8 Add an End or Terminate End to the subprocess definition.

   **TIP** The End element ends the subprocess defined by the workflow path to which it is added, but it allows its subprocesses to complete. The Terminate End element terminates the entire subprocess and all of its subprocesses.

9 Add sequence flows to connect the elements in the subprocess. To connect two elements together, click \(
\text{link mode symbol} \) to enable link mode, and then drag the cursor from the first element to the next.

   **TIP** To move elements in the diagram, click \( \text{move mode symbol} \) to enable move mode. You can switch between link mode (\( \text{link mode symbol} \)) and move mode (\( \text{move mode symbol} \)) as needed.

   For more information, see “Add Sequence Flows” on page 12.

10 (Optional) Specify conditions for the sequence flows. The conditions on sequence flows are evaluated when a workflow instance executes, and the results determine which paths are taken. See “Add Sequence Flow Conditions” on page 13 for more information.

11 Click \( \text{save symbol} \) to save the workflow definition.
Add a Call Activity

1. Drag the **Call Activity** from the list of objects onto the workflow diagram.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Click to display the **Process** tab.

4. Select the process (workflow) definition that you want to call.

5. (Optional) Specify the input parameters for the call activity. Input parameters pass data object values into the call activity. Select the data objects in the current workflow (source) whose values need to be passed to the call activity process (target).
   a. Click above the **Input Parameters** table. The New Input Parameter window appears.
   b. In the **Source** field, select the data object whose value you want to pass to the call activity.
   c. In the **Target** field, select the data object in which you want to store the value of the source data object.
   d. Click **OK** to close the New Input Parameter window.

   **TIP** To edit or remove an input parameter, select the parameter in the table and click or .

6. (Optional) Specify the output parameters for the call activity. Output parameters pass data object values from the call activity.
   a. Click above the **Output Parameters** table. The New Output Parameter window appears.
   b. In the **Source** field, select the data object whose value you want to pass to the current workflow.
   c. In the **Target** field, select the data object in which you want to store the value of the source data object.
   d. Click **OK** to close the New Output Parameter window.

   **TIP** To edit or remove an output parameter, select the parameter and click or .
Using Timers

About Timers

You can use timers to trigger workflow activities at specific times or after specified intervals. For example, you can use timers to delay the start of a task or subprocess, cancel an activity after a specified interval, or send an email reminder at a specific time.

SAS Workflow Manager provides two different types of timers:

**boundary timers**

are associated with a specific activity. The timer starts when process execution arrives at the associated activity. The timer can be set to fire immediately, after a relative delay, or at a specific time. You can set boundary timers to repeat at specific intervals, or you can use them to cancel the associated activity. When the timer fires, all of the outgoing paths from the timer are followed. If the timer is set to repeat, all outgoing paths are followed every time the timer fires. If the associated activity completes before the timer fires, the timer is canceled.

**Note:** A boundary timer must have at least one outgoing sequence flow.

**intermediate timers**

are independent elements of a workflow. They are not attached to a specific activity. The timer starts when process execution arrives at the timer. The timer can be set to fire immediately, after a relative delay, or at a specific time. Intermediate timers cannot be set to repeat, and they cannot be used to cancel an activity. When the timer fires, all of the outgoing paths from the timer are followed.

If you set a timer to fire at a specific time and that time has already passed when the timer starts, then the timer fires immediately.

**Note:** Datetime type data objects without default values are initialized to the current datetime at process start. Ensure that the default values for Datetime type data objects are set properly before the timer is accessed.
Add a Boundary Timer

1. Drag the **Boundary Timer** from the list of elements onto the task with which you want to associate the timer.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Click to display the **Timer** property pane.

4. (Optional) Clear the **Cancel the activity** check box on the **First Occurrence** tab if you do not want the timer to cancel the activity when it fires.

5. Set the time at which you want the timer to fire.
   - If you want the timer to fire immediately when the workflow reaches the activity, select **When the timer is reached**.
   - If you want the timer to fire after a delay, select the delay under **Add this delay**. This delay is relative to the time at which the timer started.
   - If you want to exclude specific dates for the boundary timer, you can add a business calendar. Select **Dates to exclude** and select the business calendar from the drop-down list. Click **View item** to view the selected calendar.

   **Note:** Only a SAS Workflow Administrator can create, edit, or delete business calendars. For more information, see “Managing Business Calendars” in SAS Workflow Manager: Administrator’s Guide.

   If you want the timer to fire at a specific time, select **At the date and time specified by this data object**, and select the data object of type DATETIME. The data object is evaluated when the associated task starts.

   **Note:** If you set a timer to fire at a specific time and that time has already passed when the timer starts, then the timer fires immediately.

6. (Optional) Set the timer to repeat.
   - Clear the **This event occurs only once** check box on the **Recurrence** tab.

     **Note:** You must clear the **Cancel the activity** check box in order to perform this action.

   - Specify the recurrence interval.

   c. If you want to exclude specific dates for the recurring boundary timer, you can add a business calendar. Select **Dates to exclude** and select the business calendar from the drop-down list. Click **View item** to view the selected calendar.
Add an Intermediate Timer

1 Drag the Intermediate Timer from the list of elements onto the workflow diagram.

2 (Optional) On the Properties pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3 Click ☰ to display the Timer property pane.

4 Set the time at which you want the timer to fire.
   - If you want the timer to fire immediately when the workflow reaches the activity, select When the timer is reached.
   - If you want the timer to fire after a delay, select the delay under Add this delay. This delay is relative to the time at which the timer started.
   - If you want the timer to fire at a specific time, select At the date and time specified by this data object, and select the data object of type DATETIME. The data object is evaluated when the instance that contains the timer starts.

TIP You can also highlight the element and click ☰ in the properties pane. Click ☰ above the boundary timer table to configure a timer.

TIP You can also click  on the element or right-click the element to add a new boundary timer by using the current elements pop-up menu.
Note: If you set a timer to fire at a specific time and that time has already passed when the timer starts, then the timer fires immediately.

TIP You can also click : on the element or right-click the element to add a new intermediate timer using the current elements pop-up menu.
Using Messages

About Messages

Messages provide a way for applications to communicate with the workflow. Messages are character strings that are sent to a running workflow instance by another process. The process sends the message through the workflow service.

You can use messages to trigger workflow activities and to cancel activities.

SAS Workflow Manager provides two different types of messages:

boundary messages
are associated with a specific activity. You can use boundary messages to cancel the associated activity. When process execution arrives at the associated activity, the workflow waits for the message to arrive before either proceeding with the process or canceling the activity. When the message arrives, all of the outgoing paths from the message event are followed. If the associated activity completes before the message arrives, the message event is canceled.

intermediate messages
are independent elements of a workflow. They are not attached to a specific activity. When process execution arrives at the intermediate message event, the workflow waits for the message to arrive. When the message arrives, all of the outgoing paths from the message event are taken.

Note: A message reference is required for boundary messages. You can create a boundary message without a message reference value, but the workflow definition fails when you try to create a new version. A workflow definition is not fully validated until it is saved as a version.
Add a Boundary Message

1. Drag the **Boundary Message** from the list of objects onto the task with which you want to associate the message.

   **TIP**  You can also highlight the activity and click in the properties pane. Click in the **Boundary Messages** property pane to configure a new boundary message.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Click to display the **Boundary Messages** property pane.

4. (Optional) Clear the **Cancel the activity** check box if you do not want to cancel the activity when the message arrives.

5. Enter the **Message reference**. The message reference is the label used to identify the appropriate character string that is sent to the running workflow.

   **TIP**  When you add a boundary message, the property pane automatically displays the properties for the boundary message. To delete or edit the properties of a boundary message later, right-click the boundary message, and select **Edit** or **Delete**.

   You can also highlight the activity and click to access the properties pane. Select the boundary message, and click to edit or to remove.

---

Add an Intermediate Message

1. Drag the **Intermediate Message** from the list of objects onto the workflow diagram.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Click to display the message properties pane.

4. Enter the **Message reference**. The message reference is the label used to identify the appropriate character string that is sent to the running workflow.

   **TIP**  To edit the properties of an intermediate message later, click the intermediate message. The property pane automatically displays the
properties for the intermediate message, and you can edit the message reference.

To delete an intermediate message later, right-click the intermediate message and select **Delete**.
About Gateways

Gateways enable you to control or synchronize how sequence flows converge and diverge in a workflow. For example, your workflow might have five workflow paths that converge to a gateway that has two outgoing paths, and the outgoing paths must all be taken at the same time. You can use a parallel gateway to control this flow.

If you do not need to control how paths diverge and converge, then you do not need to use a gateway. For more information about controlling the flow of a workflow, see Chapter 10, “Understanding Workflow Patterns,” on page 71.

Table 8.1 on page 63 describes the gateway types provided by SAS Workflow Manager.

Table 8.1 Gateway Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Action when used to fork paths</th>
<th>Action when used to merge paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>The outgoing paths. You cannot specify conditions for outgoing paths on a parallel gateway. For more information, see “Using Parallel Gateways (AND-split)” on page 73.</td>
<td>The workflow waits for all of the incoming paths to arrive before it allows the workflow process to continue executing. If any incoming path is not active, the workflow becomes deadlocked. For more information, see “Merging with Parallel Gateways (Join)” on page 79.</td>
</tr>
</tbody>
</table>
**Type** | **Action when used to fork paths** | **Action when used to merge paths**
--- | --- | ---
Exclusiv e | The workflow takes only one outgoing path. The path that is taken is the first path whose condition evaluates to True. For more information, see “Using Exclusive Gateways (XOR)” on page 74. | You should not use exclusive gateways to merge paths. When an exclusive gateway is used to merge paths, it behaves like a simple merge. See “Merging Alternate Paths (Simple Merge)” on page 78 for more information.
Inclusiv e | The workflow either takes all of the outgoing sequence flows whose conditions evaluate to True, or it takes the default path if none evaluate to True and you have designated a default path. For more information, see “Using Inclusive Gateways (OR-split)” on page 75. | The workflow waits for all of the active incoming paths to arrive before it allows the workflow process to continue executing. If at least one of the incoming paths is active, the gateway allows the workflow process to continue. For more information, see “Merging with Inclusive Gateways (Inclusive Merge)” on page 79.
Event-based | The workflow takes only one outgoing path. The path that is taken is the first path that is triggered by an event. For more information, see “Using Event-Based Gateways” on page 77. | You should not use event-based gateways to merge paths.

### Add a Parallel Gateway

1. Drag the **Parallel Gateway** object from the list of elements onto the workflow diagram.

2. (Optional) On the **Properties** pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Click to enable link mode, and connect the appropriate workflow elements to the gateway.

### Add an Exclusive Gateway

1. Drag the **Exclusive Gateway** object from the list of elements onto the workflow diagram.
2 (Optional) On the Properties pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3 Click to enable link mode, and connect the appropriate workflow elements to the gateway.

4 Enter conditional expressions for the outgoing sequence flows. For each sequence flow that requires a conditional expression, complete the steps listed in “Add Sequence Flow Conditions” on page 13.

5 Select the gateway.

6 Click to display the Conditions property pane.

7 Verify that the conditional expressions are listed in the order in which you want them to be evaluated when the workflow is running. For more information, see “Using Inclusive Gateways (OR-split)” on page 75.

---

Add an Inclusive Gateway

1 Drag the Inclusive Gateway object from the list of objects onto the workflow diagram.

2 (Optional) On the Properties pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3 Click to enable link mode, and connect the appropriate workflow elements to the gateway.

4 Enter conditional expressions for the outgoing sequence flows. For each sequence flow that requires a conditional expression, complete the steps listed in “Add Sequence Flow Conditions” on page 13.

5 Select the gateway.

6 Click to display the Conditions property pane.

7 Verify that the conditional expressions are listed in the order in which you want them to be evaluated when the workflow is running. For more information, see “Using Inclusive Gateways (OR-split)” on page 75.

8 (Optional) To designate a default sequence flow, select the row for the sequence flow, and click . SAS Workflow Manager displays a check mark (✓) next to the default sequence flow. For more information, see “Using Inclusive Gateways (OR-split)” on page 75.

**TIP** To remove the default designation, select the default and click .
Add an Event-Based Gateway

1. Drag the Event-based Gateway object from the list of elements onto the workflow diagram.

2. (Optional) On the Properties pane, edit the name and description. See “Edit Element Properties” on page 11 for more information.

3. Drag either an Intermediate Message or Intermediate Timer element onto the workflow diagram.

4. Click \( \text{\texttrade} \) to enable link mode, and connect the appropriate workflow elements to the gateway.

5. Enter conditional expressions for the outgoing sequence flows. For each sequence flow that requires a conditional expression, complete the steps listed in “Add Sequence Flow Conditions” on page 13.

6. Select the gateway.

7. Click \( \text{\texttrade} \) to display the Conditions property pane.

8. Verify that the conditional expressions are listed in the order in which you want them to be evaluated when the workflow is running.
Localizing Workflow Text

About Languages and Workflows

You can specify localized text for the following items in a workflow definition:

- names and descriptions of definitions, subprocesses, user tasks, and service tasks
- names of data objects
- display text for prompts

You can also specify a default language for use when you are working with a workflow definition in SAS Workflow Manager.

Localized Text in SAS Workflow Manager

When you are defining a workflow in SAS Workflow Manager, the text that you see is determined as follows:

1. If localized text is defined for your current locale, you see text in the language for your locale.
2. For any element that does not have localized text for your current locale, you see the localized text for the default language that is specified in the definition.
If an element does not have localized text for the default language specified in the workflow definition, you see English text for the \texttt{en-US} locale.

\textbf{Note:} Date and number formats are localized for the locale that was in use when the workflow definition was created. If you import the definition into an environment that is using a different locale, the localized date and number formats remain localized for the original locale.

\section*{Localized Text in a Workflow Instance}

For users that are working with a running instance of a workflow, the text that the users see is determined by the locale that is specified in the \texttt{ACCEPT-LANGUAGE} header field. This header field is in the request that is sent to the workflow service by the client application.

1. If localized text is defined for the locale that is specified in the \texttt{ACCEPT-LANGUAGE} header, the user sees the text for that locale.

2. For any element that does not have localized text for that locale, the user sees the text that was entered into SAS Workflow Manager when the workflow was defined.

\section*{Set the Default Language}

\textbf{TIP} The default language specified in the workflow definition is used only in SAS Workflow Manager. The default language does not affect what the user sees in a running workflow instance.

1. Click the tab for the main process.

2. Click \textquoteleft in the object toolbar, and select \texttt{Languages}. The \texttt{Localize Text} window appears.

3. Select the default language, and click \texttt{OK}.

\section*{Localize Definition Names and Descriptions}

1. Click the tab for the main process.

2. Click \textquoteleft, and select \texttt{Languages}. The \texttt{Localize Text} window appears.
Complete these steps for each language:

a. Click above the Name table. The Localized Text window appears.

b. Select the language.

c. Enter the definition name in the selected language in the text field.

d. Click OK.

4 Repeat Step 3 for the Description.

TIP Only localized definition names and descriptions that are not in the set default language can be edited or removed from the localized text window. Select the definition name or description, click to edit or to remove.

Localize Subprocess, User Task, and Service Task Information

1 Select the element for which you want to localize the name and description.

2 Click to display the Localization property pane.

3 Complete these steps for each language:

a. Click above the Name table. The Localized Text window appears.

b. Select the language.

c. Enter the element name in the selected language in the text field.

d. Click OK.

4 Repeat Step 3 for the Description.

TIP Only localized definition names and descriptions that are not in the set default language can be edited or removed from the localized text window. Select the definition name or description, click to edit or to remove.

Localize Data Object Names

1 Click the tab for the main process or a subprocess, depending on whether you want to localize global or local data object names. See “About Data Objects and Scope” on page 17 for more information.
2 Click **. The Definition Properties window appears.

3 For each data object, complete these steps for each language:
   a Select the data object and click **Localize**. The Localize Text window appears.
   b Click **. The Localized Text window appears.
   c Select the language.
   d Enter the data object name in the selected language in the text field.
   e Click **OK** to close the Localized Text window.

4 Click **OK** to close the Definition Properties window.

**TIP** Only localized data objects that are not in the set default language can be edited or removed from the localized text window. Select the data object, click ** to edit or ** to remove.

---

**Localyze Prompts**

1 Select the user task or Start element for which you want to localize the prompts.

2 Click ** to display the **Prompts** property pane.

3 For each prompt, complete these steps for each language:
   a Select the prompt and click **Localize**. The Localize Text window appears.
   b Click **. The Localized Text window appears.
   c Select the language.
   d Enter the prompt text in the selected language in the text field.
   e Click **OK** to close the Localized Text window.

**TIP** Only localized prompts that are not in the set default language can be edited or removed from the localized text window. Select the prompt, click ** to edit or ** to remove.
About Workflow Patterns

Only the most basic workflows can be represented as a single, sequential flow. Most workflows contain combinations of activities, events, and gateways. A workflow definition defines the possible paths through these elements and is based on your organization's business processes. The actual path that is taken by a workflow instance is determined at run time by the outcome of each activity and by the evaluation of conditional expressions associated with sequence flows.

A workflow definition typically uses one or more workflow patterns. A workflow pattern is a process construct that defines how a workflow behaves. For example, many workflows have a decision point where the main workflow path diverges, and alternative paths are available through the workflow. Alternate paths might be mutually exclusive, or they might run in parallel. You can control how the alternate
paths are followed by using conditional expressions. How the workflow behaves depends on the workflow patterns that are modeled in the definition.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>The workflow defines a single path through the workflow elements.</td>
</tr>
<tr>
<td></td>
<td>See “Sequential Flow” on page 72 for more information.</td>
</tr>
<tr>
<td>Divergent</td>
<td>The workflow defines a path that splits into multiple alternative paths. If</td>
</tr>
<tr>
<td></td>
<td>more than one path is executed, they are executed in parallel. You can</td>
</tr>
<tr>
<td></td>
<td>use conditional sequence flows or gateways to implement divergent patterns.</td>
</tr>
<tr>
<td></td>
<td>See “Divergent Patterns” on page 73 for more information.</td>
</tr>
<tr>
<td>Convergent</td>
<td>The workflow defines multiple paths that merge at the same element.</td>
</tr>
<tr>
<td></td>
<td>See “Convergent Patterns” on page 78 for more information.</td>
</tr>
<tr>
<td>Termination</td>
<td>A workflow path reaches an End or Terminate End element.</td>
</tr>
<tr>
<td></td>
<td>See “Termination Patterns” on page 80 for more information.</td>
</tr>
<tr>
<td>Cancellation</td>
<td>An activity or process is canceled, and the workflow does not reach an</td>
</tr>
<tr>
<td></td>
<td>End or Terminate End element.</td>
</tr>
<tr>
<td></td>
<td>See “Cancellation Patterns” on page 82</td>
</tr>
<tr>
<td>Multiple Instance</td>
<td>A separate instance of a task is created for each participant.</td>
</tr>
<tr>
<td></td>
<td>See “Multiple Instance Pattern” on page 80 for more information.</td>
</tr>
</tbody>
</table>

**Sequential Flow**

The most basic workflows are represented as a single sequential flow. Each task is completed in sequence, and no conditions are associated with the sequence flows. However, most workflows are not this simple.

**Figure 10.1  Sequential Workflow Diagram**
Divergent Patterns

Using Alternate Paths

Some workflows have decision points and alternate paths, and each path represents a different business case. For example, a specific path might depend on the value of a data object or on a participant’s response to a prompt. Suppose you have the workflow shown in Figure 10.2 on page 73. You can define a prompt for the Start node. Then, you can define conditional expressions for the outgoing sequence flows on the Start node that evaluate the participant’s answer. Depending on the results, the running workflow might take the path to Task 1, the path to Task 3, or both paths. If it takes both paths, then Tasks 1 and 3 are executed in parallel.

If neither of the conditional expressions defined for the sequence flows on the Start node evaluate to True, the workflow process terminates.

Figure 10.2 Workflow with Alternate Paths

An alternative to simple alternate paths is an inclusive gateway. For more information, see “Using Inclusive Gateways (OR-split)” on page 75.

Using Parallel Gateways (AND-split)

You can use a parallel gateway when the workflow needs to take multiple paths at the same time. You cannot associate conditions with the outgoing sequence flows from a parallel gateway. All of the outgoing paths are followed, and the subsequent activities run in parallel. In the workflow shown in Figure 10.3, Tasks 2, 3, and 4 all run in parallel.
Using Exclusive Gateways (XOR)

With an exclusive gateway, only a single outgoing path is executed. The path that is chosen is based on the conditions that you specify for the outgoing sequence flows. The conditions are evaluated in the order in which you define them. Only the sequence flow for the first condition that evaluates to True is executed.

A sequence flow with no associated conditions becomes the default sequence flow. If the conditions on all of the other sequence flows evaluate to False, the default sequence flow is taken. When you specify the order in which the sequence flows are evaluated, the default sequence flow must be the last one in the list.

For example, suppose the workflow shown in Figure 10.3 defines an optional prompt for the task Enter Order that prompts the participant to enter their age. The conditions that are defined for outgoing sequence flows on the gateway are as follows:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>age&lt;18</td>
<td>Minors</td>
</tr>
<tr>
<td>age&gt;=18</td>
<td>Adults</td>
</tr>
<tr>
<td></td>
<td>No Response</td>
</tr>
</tbody>
</table>

If the prompt is optional and the user chooses not to enter their age, the workflow sets the value to zero and takes the path to the No Response task.
Using Inclusive Gateways (OR-split)

For an inclusive gateway, all of the sequence flows whose conditions evaluate to True are taken. A sequence flow with no associated conditions always evaluates to True, and that path is always taken. (A sequence flow with no associated condition is an unconditional sequence flow.)

For example, suppose Task 1 in Figure 10.5 on page 76 requires that the participant enter a customer’s order information. The customer could order CDs and DVDs, which require special handling. The following conditions are defined for outgoing sequence flows on the gateway:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>numCDs&gt;0</td>
<td>CDs</td>
</tr>
<tr>
<td>numDVDs&gt;0</td>
<td>DVDs</td>
</tr>
<tr>
<td></td>
<td>No CDs or DVDs</td>
</tr>
</tbody>
</table>

If a customer orders CDs, the CDs task starts. If the customer orders DVDs, the DVDs task starts. If the customer orders both CDs and DVDs, then both of those tasks start. If the customer’s order does not contain any CDs or DVDs, then only the default task (No CDs or DVDs) starts.
When you configure the gateway, you specify the order in which the sequence flows are evaluated. You can also designate a default sequence flow. The default sequence flow is taken only if all of the other paths evaluate to False. If any of the sequence flows are unconditional, you can designate one of the unconditional sequence flows only as the default flow. If all of the sequence flows have conditions, then you can designate any of the sequence flows as the default flow.

For example, suppose you have an inclusive gateway that diverges to three tasks. Only one of the outgoing sequence flows has a condition, and the first of the unconditional sequence flows is designated as the default flow. The last sequence flow in the list is unconditional. The table for this gateway is shown in Table 10.2 on page 76.

If the condition on the first sequence flow evaluates to False, this workflow takes only the path to Task 3.

Table 10.2  Inclusive Gateway Conditions That Execute Only Task 3

<table>
<thead>
<tr>
<th>Condition</th>
<th>Endpoint</th>
<th>Default</th>
<th>Path Is Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluates to False</td>
<td>Task 1</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>No condition (always True)</td>
<td>Task 2</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>No condition (always True)</td>
<td>Task 3</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For more information, see “Add an Exclusive Gateway” on page 64.
Using Event-Based Gateways

An event-based gateway determines the path that a process takes, based on events. Each outgoing sequence flow must be connected to an intermediate timer or message event. When a process reaches an event-based gateway, the execution is suspended, and for each outgoing sequence flow, an event subscription is created. The process follows the path for the event that occurs first. No other events in the gateway can be triggered after the path is determined.

For example, suppose the workflow shown in Figure 10.6 illustrates an event-based gateway when the payment on an order has failed. The process can take one of two paths. In the first path, an intermediate message event occurs and the payment details are updated. Then, the process proceeds to a service task where the payment is processed. In the second path, an intermediate timer event is triggered. If the payment details are not updated within two hours, the process proceeds to a service task where the order is canceled. Whichever intermediate event occurs first determines which path the process takes. The conditions that are defined for outgoing sequence flows on the gateway are described in the following table.

<table>
<thead>
<tr>
<th>Intermediate catching event</th>
<th>Condition</th>
<th>Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate Message</td>
<td>Payment details are updated</td>
<td>Payment is processed</td>
</tr>
<tr>
<td>Intermediate Timer</td>
<td>If there is no response after two hours, proceed with cancellation</td>
<td>Order is canceled</td>
</tr>
</tbody>
</table>

Table 10.3 Conditions and Endpoints for a Divergent Workflow with an Event-Based Gateway

Divergent Patterns
Convergent Patterns

Merging Alternate Paths (Simple Merge)

It is possible to merge workflow paths by doing a simple merge in which multiple outgoing paths merge at the same element. Simple merges do not use a gateway.

Simple merges are most appropriate when the paths are mutually exclusive. Suppose you have the workflow shown in Figure 10.7 on page 78. If the paths are mutually exclusive, the workflow takes only one of the incoming paths to Task 5, and only one instance of Task 5 is created. If the paths are not mutually exclusive and the workflow takes more than one of the incoming paths, a new instance of Task 5 is created for each path taken.

**Figure 10.7  Merging Paths without a Gateway**

![Workflow Diagram](image)

**TIP**  To run multiple occurrences of the same task, you can specify that a new instance of the task is created for each participant. See “Enabling Multiple Participants to Complete a Task” on page 30 for more information.

**Note:** A special situation exists if the paths converge at a Terminate End element. See “Explicit Termination” on page 81 for more information.
Merging with Parallel Gateways (Join)

When you merge paths by using a parallel gateway, the gateway waits for all of the incoming tasks to complete before it allows the workflow process to continue executing. (The Workflow Service knows which tasks are active and when they complete.) In Figure 10.8 on page 79, the gateway waits for Tasks 2, 3, and 4 to complete before allowing Task 5 to start.

*Figure 10.8 Convergent Workflow with a Parallel Gateway*

For more information, see "Add a Parallel Gateway" on page 64.

Merging with Inclusive Gateways (Inclusive Merge)

An inclusive gateway waits for all of the active tasks to complete. (The Workflow Service knows which tasks are active and when they complete.) For example, in Figure 10.9 on page 80 if only the paths from the CDs and DVDs tasks to the gateway are taken, the gateway triggers the Ship it! task when the CDs and DVDs tasks are complete. The workflow does not wait for the No CDs or DVDs task to run.
Multiple Instance Pattern

When you assign multiple participants to a user task, you can specify that a separate instance of the task is created for each participant. This pattern is referred to as Multiple Instances with a priori Design-Time Knowledge. For more information, see “Enabling Multiple Participants to Complete a Task” on page 30.

Note: A simple merge as described in “Merging Alternate Paths (Simple Merge)” on page 78 can create multiple instances of a task if the incoming paths are not mutually exclusive.

Termination Patterns

Termination patterns are workflow patterns in which the workflow completes normally.

Implicit Termination

The Workflow Service ends processes and subprocesses when all of their activities complete regardless of whether they encounter an End or Terminate End element. For example, suppose you have the workflow shown in Figure 10.2 on page 73. If
neither of the conditional expressions defined for the sequence flows on the Start node evaluate to True, the Workflow Service terminates the process.

However, the recommended practice is to explicitly end the paths in your workflow by adding either an End or Terminate End element to each process and subprocess.

Explicit Termination

To explicitly end workflow paths, add an End element to each path in the workflow. An End element terminates the path to which it is added, but it does not affect any other paths in workflow.

To explicitly terminate an entire process or subprocess, add a Terminate End element.

**IMPORTANT** If multiple paths merge into a Terminate End element and the paths are mutually exclusive, a single Terminate End event is triggered. If the paths are not mutually exclusive, a race condition is created. All active paths in the workflow are terminated when the first path reaches the end. The recommended practice is to use an End element. See “Implicit Termination” on page 80.

If you use a Terminate End element, you should either add mutually exclusive conditions to the paths leading to the Terminate End element or merge the paths by using a gateway.

For example, the paths that merge into the Terminate End event in the workflow shown in Figure 10.10 are mutually exclusive.

**Figure 10.10  Explicit Termination with Mutually Exclusive Paths**
Cancellation Patterns

Cancellation patterns are workflow patterns in which the workflow did not complete normally.

Canceling Activities

You can use a boundary timer or a boundary message to cancel the activity with which the timer or message is associated.

For more information, see “About Timers” on page 55 and “About Messages” on page 59.

Canceling a Workflow (Cancel Case)

Only workflow administrators can cancel a running instance of a workflow. For more information, see SAS Workflow Manager: Administrator’s Guide.

Additional Information about Workflow Patterns

For more information about workflow patterns, see the www.workflowpatterns.com website. The following documents are of particular interest:
