Active Workspace 6.0

Workflow Designer

Active Workspace 6.0
AW049 - 6.0

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit **www.siemens.com/plm**.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Managing Task Behavior

1. Overview of Workflow Designer in Active Workspace

What is Workflow Designer?

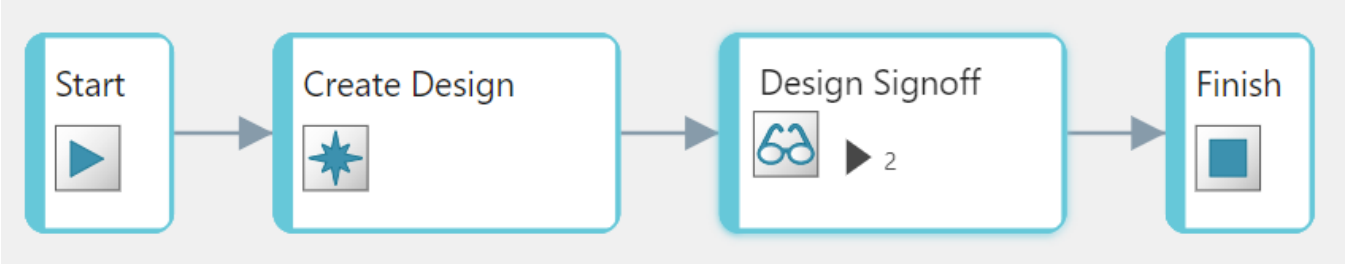
Workflow Designer lets you design workflow templates that can incorporate and automate your company's business practices and procedures. Business users then use these templates to initiate workflow processes.

The concept of a workflow is that all work goes through one or more business processes to accomplish an objective, flowing from one task to the next. Workflow is the automation of these business processes.

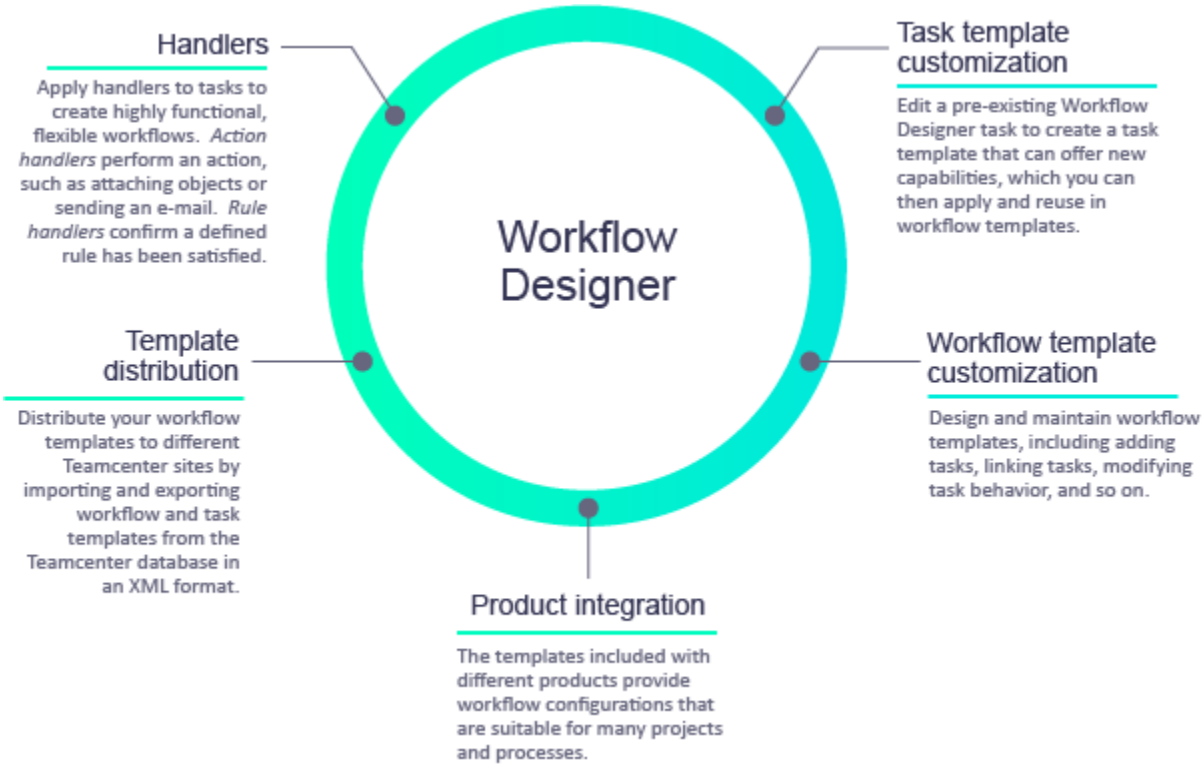
A *workflow* in Active Workspace contains a series of tasks to complete in order to reach a repeatable business goal. Customer business objects (such as documents and parts) are passed between participants during a workflow process.

Workflow *templates* define the blueprint of a workflow process to be performed at your site that can enforce best practices and eliminate errors. Templates are a good way to embed process knowledge in the project to ensure everyone on the team complies with the process, even though they may not know all its details.

In this example, the administrator created a template to request a design from a system designer (shown here by the **Do** task, **Create Design**), which is then followed by a review from other team members (the **Design Signoff** task) before it can be marked as complete (the **Finish** task). Users can reuse this template to standardize best practices and ensure that a continuous process is used throughout a project.



Workflow Designer's multiple features work together in various ways to improve your product's efficiency and productivity.



Where do I go from here?

The following table shows the high-level roles engaged with Workflow Designer.

Administrator	An **Administrator**, or a workflow template designer, in the default **DBA** group is the only user who can create templates based on existing templates or their own design. Administrators can also modify, delete, import, or export existing templates. Business users outside of the **DBA** group can be added to this workflow designer group by modifying the **Awp0IsWorkflowTemplateAuthor** condition. This condition is available in BMIDE and defines who has edit privileges. For details, see **Access templates**.
Get me started	Take a look at the prerequisites and information on how you can enable, configure, and start Workflow Designer **before you begin.**
Working with templates in Workflow Designer	• **Create a workflow template** • **Import workflow templates**

	• **Export workflow templates**
Modify an existing task to create a new one	**Create a task template**
Manage task behavior	**Modify tasks in Workflow Designer**
Learn more about workflow handlers	**Work with task handlers**
Business User	All non-administrative users (or those who do not design templates) can access the **Workflow Designer** tile to view the list of templates and what they contain in read-only mode. If non-DBA users select a workflow template, they can view its task data but can neither modify nor import or export the template. Non-DBA users also cannot see templates that are under construction.
Get me started with applying templates to workflows.	See Workflows and Tasks to learn more about applying templates to a workflow.

Before you begin

Prerequisites	Ensure that you have administrator privileges to use the Workflow Designer in **Edit** mode.
Enable Workflow Designer	To enable the Workflow Designer feature, select it during installation. If you have trouble accessing Workflow Designer, contact your system administrator.
Configure Workflow Designer	You can use Workflow Designer's default configuration settings or modify them using workflow preferences.
Start Workflow Designer	You can access the Workflow Designer tile on the home page of Active Workspace.

What's different between Workflow Designer in Active Workspace and rich client?

The functionality of Workflow Designer in Active Workspace is built upon Teamcenter Foundation, with a focus on user-centric actions. There are some functional variations between Active Workspace and the rich client.

- In the rich client, you can create a query for a Condition task. While these queries are visible in Active Workspace, you cannot create a new query here. However, you can configure an existing query that has already been created in the rich client.

- In the rich client, you create an access control list (ACL) using the task's **Attributes** panel. While these ACLs are visible in Active Workspace, you cannot create a new ACL here. However, you can configure an existing ACL that has already been created in the rich client.

What can I do with Workflow Designer?

Workflow Designer provides a graphical editor to perform the following capabilities:

- Create a workflow template, either new or based on an existing template.
- Update the workflow template to add tasks.
- Provide visual cues for process and task templates as well as online and offline templates.
- Import and export process templates, including those created in the Teamcenter rich client.
- Create custom workflow templates, update template properties, and so on, using workflow handlers. Hints are provided for the workflow handler arguments and their respective values to help users identify mandatory and optional parameters. These hints can avoid runtime errors because validation is performed during creation. A JSON file must be created for each handler to enable this feature.

Workflow templates

A *workflow template* is a predefined workflow structure that you use as a pattern for your own workflow processes. Each instance of a workflow process uses a workflow template. A workflow template sets up the rules for a workflow and is used as a blueprint for creating multiple workflow processes.

Using Workflow Designer, you can do the following:

- Create both sequential (serially) or asynchronous (in parallel) workflow process templates.
- Build new workflow process templates based on core templates.
- Define a specific workflow process by placing workflow tasks in the required performance order.
- Define additional workflow process requirements (such as placing a status on targets and creating subprocesses) in the template using workflow handlers.

Access templates

In Workflow Designer, user access to templates differs according to predefined groups, as determined by the system administrator.

- **Workflow Designers**
  System administrators in the default **DBA** group are the only users who can create templates based on existing templates or their own design. Administrators can also modify existing templates and import and export templates.
  Non-DBA can be added to this Workflow Designers group by modifying the **Awp0IsWorkflowTemplateAuthor** condition. This condition is available in BMIDE and defines who

has edit privileges. These workflow commands use server-side conditions; you can change this behavior by overriding the commands.
The Workflow Designer tile is found in the following workspaces for DBA users: **Active Admin**, **Active Architect**, and the **Default** workspace.

- **Non-Workflow Designers**
  All non-DBA users can access the Workflow Designer tile to view the list of templates and what they contain in read-only mode. If non-DBA users select a workflow template, they can view its task data but can neither modify nor import or export the template. Non-DBA users also cannot see templates that are under construction.

Workflow elements

In Teamcenter, workflows are *processes* based on *process templates* that are composed of *tasks*.

Term/Concept	Description
Workflow process	A *workflow process* is the automation of business procedures in which documents, information or tasks are passed from one participant to another in a way that is governed by rules or procedures. Teamcenter workflows allow you to manage your product data processes.
Parent process	Workflow processes can contain child workflow processes. When child workflow processes (also known as subprocesses) are created, the parent workflow processes can be dependent or independent of subprocesses.
Subprocess	Child workflow processes of a parent workflow process.
Workflow template	Blueprints of workflow processes. Your administrator creates process templates. A specific process is defined by placing tasks in the template in the required order of performance. Additional requirements, such as quorums and duration times, may also be included in the template.
Tasks	The fundamental building block used to construct a process is a task. Each task defines a set of actions, rules, and resources used to accomplish that task. User actions cause tasks to move from one state to another, and as a result, the overall process moves forward or backward.
Root task	The top-level task of every workflow is referred to as the *root task*. The root task is the top-level parent task that contains all the other tasks as subtasks.
Subtask	Child task of a parent task.
Container tasks	Tasks that contain other include tasks: • **Review** Contains **select-signoff-team** and **perform-signoffs** tasks. The **Decision** options are **Approve**, **Reject**, and **No Decision**.

Term/Concept	Description
	• **Acknowledge** Contains **select-signoff-team** and **perform-signoffs** tasks. The **Decision** options are **Acknowledged** and **Not Acknowledged**. • **Route** Contains **Review**, **Acknowledge** and **Notify** tasks. • **Task** Use it as a starting point for creating your own custom tasks, such as tasks to carry your custom forms or other site-specific tasks that the users must complete.
Interactive tasks	Tasks that require user interaction display in the affected user's worklists. Different types of tasks have different interactive requirements. Typical tasks include: • **select-signoff-team** The assigned user is required to select a signoff team to sign off the target object of the task. • **perform-signoffs** Assigned users are required to review and sign off the target object of the task. • **Do** The assigned user is required to review and perform the task instructions, then mark the task complete. • **Route** Uses the **Review**, **Acknowledge**, and **Notify** subtasks, each of which has its own dialog box. • **Task** Use it as a starting point for creating your own custom tasks, such as tasks to carry your custom forms or other site-specific tasks that the users must complete.
Workflow handlers	Small ITK programs used to extend and customize workflow tasks. Action handlers perform actions, such as attaching objects and sending email; rule handlers can identify whether a rule has been satisfied.
Task attributes	Attributes that further configure task behavior. You can set security attributes, customize task symbols, and define condition results.
Quorum requirements	Values that specify the number of approvals required before **perform-signoffs** tasks can complete and workflows can proceed.

In a workflow, actions are assigned or allowed depending on the type of user.

User	Description
Responsible party	A *responsible party* is the user responsible for performing a particular task within a workflow process. While performing the task, the responsible party can reassign responsibility of the task to another user.
Privileged user	A *privileged user* is the responsible party, the process owner, or a member of a system administration group.
Current User	The current user is the user that completed the most recent task action and is independent of the responsible party. When using the "$User" keyword in an argument, the "current user" is determined by evaluating the most recent task at the same level as the task containing the argument using the "$User" keyword. For example, if the most recent task is in a sub-process and not on the same "process level" as the task using the argument with the "$User" keyword, the argument will pull the current user from the most recent task of the parent process instead of the sub-task
Process owner	A *process owner* is the user who initiated the workflow process. The process owner is also known as the process initiator. When the workflow process is initiated, the process owner becomes the responsible party for the workflow process; the root task of the workflow process is placed in the process owner's worklist. Whenever any task in the workflow process is not explicitly assigned to another user, person or resource pool, the responsible party for the task defaults to the process owner.
Task Owner	User who is granted privileges for the task's target data.

The Workflow Designer interface

Once open, a filter results window displays the number of online and offline templates available.

The **Templates** panel is on the left. A navigation bar contains commands specific to the **Templates** panel.

Command	Description
View	Provides various display modes for the available templates: **List with Summary**, **List**, **Table**, **Table with Summary**, or **Images**.
Selection Mode	Lets you select a specific template(s).
Select All	Lets you select all available templates.

Command	Description
Filters	Lets you apply various criteria to restrict the list of available templates.
Search	Lets you search for a template by entering the name in the search field and clicking the search icon.
Open	Opens the template.

Once a workflow template is selected or opens, additional functionality is available. Additional icons appear when in edit mode.

Command	Description
Show Task Information	Displays details for the selected template under the **Properties**, **Access**, **Handlers**, **Assignments**, **Notifications**, and **Forms** tabs.
Diagram Settings	Click the **Show Grid** check box to display major or minor grid lines to assist with symmetrical layout.
Apply Layout	Provides **various options** to display the task order.
Task Palette	Displays the various task types to add to the workflow template, as well the success or fail path branches, when the template is in edit mode.
Remove	Deletes an object from the graphic editor when the object is selected in edit mode.
Save Layout	Saves the current layout of the workflow and overwrites the position values displayed in either the fixed layout mode or Workflow Designer in the Teamcenter rich client. This command is only shown in the auto layout mode. This is because the fixed layout mode automatically saves the task node positions as they are moved. The preference **WRKFLW_save_node_positions** must be set to **true** for this option to display.
Online	Makes the template online and available for users. Once the edits are saved and you place the template online, the newly edited template overwrites the previous version.
Start Edit	Allows the user to edit the selected template. You can choose whether the template should be kept online or offline while it is being edited.

Command	Description
End Edit	Stops the template editing process and saves any updates.
Full Screen	Takes over the full browser window for the display.

The Primary toolbar contains additional Workflow Designer functionality.

Command	Description
New	Provides the options to create or import a new workflow template. The menu options under **New** are not displayed if a template is already open.
Create Workflow Templates	Creates a template, providing the option to enter a name, description, and whether the template is based on one that already exists.
Import	Imports an existing template from a file outside of Workflow Designer.
Export	The **Export** option exports a template to an .XML file outside of Workflow Designer. You can use multi-select to export multiple templates at once. Note: The **Export** option is available under the **Share** menu.
Edit	Displays the editing options.
Start Edit	Allows the user to edit the selected template. You can choose whether the template should be kept online or offline while it is being edited. Template edits are saved automatically as you make them.
Offline	Provides the user the option of keeping the template offline while being edited. In this case, you see only one instance of the template, which shows the under construction check mark icon. The template is not available to other users while you are editing it.
Online	Makes the template online and available for users. Once the edits are saved and you place the template online, the newly edited template overwrites the previous version.
End Edit	Stops the template editing process and saves any updates.
Delete	Deletes the workflow process template.

See **Creating Workflow Templates** for more information.

2. Creating workflow templates in Active Workspace

Workflow templates overview

Basic workflow template process

A *workflow* describes the individual tasks and the task sequence required to model the workflow process.

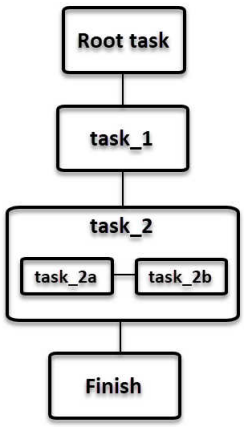
A *workflow template* is a predefined workflow structure that you can use as a pattern for your own workflows. You can define a specific workflow by placing workflow tasks in the order required. You can define additional workflow process requirements, such as placing a status on targets,and creating subworkflows (also known as subprocesses).

Workflow templates are created from scratch or based on existing templates. Existing templates include the out-of-the-box default templates included with Workflow Designer or those created by your organization.

You can use the out-of-the-box templates as a basis for creating your own templates, configured to include specific assignments, automatic notifications, or a unique purpose of the workflow.

> Note:
>
> The out-of-the-box templates will always override during upgrades, so if you want to isolate your configuration changes, create Saved As copies of the templates. You can always view the latest changes on the out-of-the-box templates and then decide which changes you want to adopt at your own pace.

Each workflow contains a group of nested tasks. The top-level task of every workflow is referred to as the *root task*, as shown here in an example workflow structure.



The root task is the top-level parent task that contains all the other tasks as subtasks. It is the first task run when a workflow process is initiated and the last task to complete before the workflow itself is completed.

In a relatively simple workflow, as shown in the following example, the root task (Start) leads to a **Do** task (Complete the work). The **Do** task leads to a **Review** task (Review the work, which contains the **select-signoff-team** and **perform-signoffs** subtasks), which then leads to an **Acknowledge** task (Acknowledge the work, which also contains the **select-signoff-team** and **perform-signoffs** subtasks) and finally to a **Finish** step.

To design and maintain workflow processes in Workflow Designer, you can perform the following actions:

- Create templates
- View templates
- Add tasks to templates
- Link tasks
- Modify task behavior
- Import and export workflow templates

Create a workflow template

1. Click **New** > **Create Workflow Template**.
2. Click **Workflow** as the **Template Type**.
3. Enter a **Name** and **Description**. A unique name is required. Entering a duplicate name will display an error message.
4. Under **Based On**, select an existing template on which to base the new task, or select **None** to create your own template.

5. Click **Add** to create the new template.

   A basic template contains the start and end tasks. Add the individual tasks and paths to customize the template.

Add tasks to a template

After creating a template, you can add tasks.

Add tasks to a template

1. Select your newly created template from the list.

   The initial template only contains the start and end tasks.

2. Note that the initial creation of a template automatically puts it in edit mode. However, if you had created a template, left it, and then came back to it, click **Edit** > **Start Edit** to make any changes.

3. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

4. (Optional) Click **Settings** to show the layout grid for aligning your tasks.

5. Click **Task Palette** to add tasks to the template.

6. Select a task and drag to the template. The task name is highlighted for editing.

   Continue adding tasks in the same manner.

7. Click **End Edit** to save your new task(s) to the template.

Add paths between tasks

After adding one or more tasks, add paths connecting the tasks promoting the Subworkflows progression.

There are two basic path types, **Success** or **Fail**.

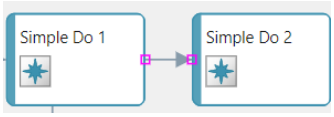
1. Click **Task Palette** to add paths to the template.

2. On the **Task Palette**, select a **Success** or **Fail** path.

3. Move your cursor over a task. A cross-hair indicates the starting point for the path. Click and hold the cursor to draw a path between tasks. Release the cursor to view the new path.

> Tip:
>
> If you select a path, you can draw all the same path types at one time. For instance, select **Success**, and draw all of the **Success** paths followed by all of the **Fail** paths.

4. To change the established path type, click the path to display the pink line indicators, and click **Remove** ⊖. Then select **Success** or **Fail** from the **Task Palette**.



5. Click **Edit** ✎ > **Online**.

   The template is now available to the organization.

Task template types

A *task template* is a blueprint of a workflow task. A task is a fundamental building block used to construct a workflow process. Each task defines a set of actions, rules, and resources used to accomplish that task.

Task templates can be both **designed by yourself** to apply and reuse in workflow templates or provided by the Workflow Designer, as shown below.

Symbol	Task	Definition
	Acknowledge Task	Uses the **Acknowledged** and **Not Acknowledged** subtasks, each of which has its own dialog.
	Add Status Task	Creates and adds a release status to the target objects of the workflow process. It is a visual milestone in a workflow process. No dialog box is associated with this type of task.
	Condition Task	Branches a workflow according to the defined query criteria. It requires that the succeeding task containing an **EPM-check-condition** handler that accepts a Boolean value of either **True** or **False**.
	Do Task	Allows the user to choose from two options if at least one failure path is configured. **Complete** confirms the completion of a task and triggers the branching to a success path. **Unable to Complete** indicates the task could not be completed, for varying reasons.

Symbol	Task	Definition
	**Or Task**	Continues the workflow process when any one of its multiple task predecessors is completed or promoted. There is no limit to the number of predecessors an **Or** task may have.
	**Review Task**	Uses the **select-signoff-team** and **perform-signoffs** subtasks, each of which has its own dialog. **Full Participation Required** is an option that allows the user to set the **Review** task to wait for all reviewers to submit their decisions before completing and following the appropriate path.
	**Route Task**	Uses the **Review**, **Acknowledge**, and **Notify** subtasks, each of which has its own dialog box.
	**Task**	Use it as a starting point for creating your own custom tasks, such as tasks to carry your custom forms or other site-specific tasks that the users must complete.
	**Validate Task**	Branches a workflow along two or more paths. Active paths flowing out of the task are determined by whether specified workflow errors occur. Use this task to design workflows around anticipated errors.

Customize task paths

There are two basic path types between tasks **Success** or **Fail**.

You can modify **Success** paths for **Condition** tasks adding a customized response, instead of the default **True** or **False** values. You cannot modify fail paths.

1. Click **Task Palette** to add a **Success** path from a **Condition** task.

2. Close the **Task Palette** and select the path.

   The **Properties** tab displays the specific options relative to the selected item. For **Success** paths from a **Condition** task, the default Properties are **true** and **false**.

3. Enter a new value in the **Add Custom Result** box and click **Add**.

   The new result is added to the **Results List**.

4. In the **Results List**, click the list and clear the value that is no longer desired, for instance, **True**.

   The new path value is reflected in the workflow template.

Subworkflows

What are subworkflows?

Sometimes you want a workflow process to generate additional workflows as it proceeds. For example, you may want a workflow to generate additional workflows for each target of the parent workflow. This is useful if you want each target to undergo a separate review and signoff process.

Subworkflows (or subprocesses) are child workflows of a parent workflow. For example, users might want to create a subworkflow after receiving a task in their worklist dependent upon the completion of one or more tasks not tracked by the existing workflow. They create a subworkflow to track the additional tasks. In other cases, when working on a task, a user might discover that the opinion of another user would be useful. Creating an ad hoc subworkflow includes that second user as part of the existing workflow.

Subworkflows are created in two locations:

Parent workflow templates	Administrators can configure workflow templates to create subworkflows. For example, a parent workflow template can be configured to automatically launch subworkflows for each target of the parent workflow via the **EPM-create-sub-process** handler (see Create a Subworkflow for details).
My Worklist	Generally, any user can create an ad hoc subworkflow from a task within their worklist. This functionality is not limited to privileged users. When you create a subworkflow from an in-process task in your worklist, you create a dependency between the selected task in the parent process and the newly created subworkflow. If a subworkflow is created from an in-process task, the task cannot be completed until the subworkflow is completed. Regardless of how these two preferences are set to control the inheritance of target objects from the parent workflow, users can always manually remove targets from subworkflows.

Consider the following when working with subworkflows:

If the parent workflow is dependent on the subworkflow, the parent workflow cannot be completed until the subworkflow is completed.

For example, if the **EPM-create-sub-process** action handler is used to create subworkflows for multiple targets from a parent workflow, the parent workflows are dependent on the subworkflows.

Access to create subworkflows is governed by the Access Manager **Has Class (Task)** rule and the **Task Named** access control list (ACL). The same permissions allowing you to perform the task allow you to create a subworkflow from the task.

Workflow subprocesses can be dependent or independent of parent processes, depending on the preference.

Independent of parent process

The **WRKFLW_skip_abort_on_sub_process** preference is honored only for independent subprocesses.

Set the **WRKFLW_skip_abort_on_sub_process** preference to true to skip abort of subprocess when a parent process is aborted.

Dependent upon parent process

If there is a dependency from a parent process to its subprocesses, aborting the parent will abort the dependent subprocesses, irrespective of the value of the **WRKFLW_skip_abort_on_sub_process** preference.

The default value of **WRKFLW_skip_abort_on_sub_process** is false which will abort the subprocesses along with parent process.

The **WRKFLW_parent_behavior_on_sub_process_abort** preference ensures the parent process's task proceeds after the dependent subprocess is aborted.

Create a subworkflow

The **EPM-create-sub-process** handler starts subworkflows (also known as subprocesses) from a workflow process. The new subworkflow can take on attachments of the parent process, and these attachments can be grouped by property.

To create a subworkflow using this handler:

1. Select a template from the list.
2. Click **Edit** > **Start Edit**.
3. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.
4. Click **Task Palette**.
5. Select the task to contain the subworkflow.
6. Click the **Handlers** tab.

7. Under **Action Handlers**, click **Add** ⊕.

8. From the **Handler** list, select the **EPM-create-sub-process** handler.

9. Add the name of the template to contain the subworkflow in the **Value** field.

10. Click **Add**.

11. Click **End Edit** to save your new subworkflow to the template.

You can add the **EPM-create-sub-process** handler multiple times to a single task, allowing you to use different workflow templates per target object type. Use the handler to:

- Set dependencies between the parent workflow and its subworkflows.
- Define targets and attachments for the subworkflows.
- Transfer attachments from the parent workflow to a subworkflow.
- Create subworkflows for multiple targets.
- Create subworkflows for assemblies.
- Create subworkflows for related objects.

The handler accepts numerous arguments, allowing you to create a wide variety of instances for generating subworkflows. For example:

- The following argument settings create a subworkflow based on the **Clinical Trials Phase I** template, which inherits all the targets and reference attachments from the parent workflow. Because the workflow name is not defined, a workflow name for the child workflow is automatically generated in the format *parentprocess:count*.

Argument	Value
-template	**Clinical Trials Phase I**
-from_attach	**ALL**
-to_attach	**ALL**

- The following argument settings launch a subworkflow based on the **Clinical Trials Phase I** workflow template. All item revisions from the parent workflow are excluded as targets for the new workflow.

Argument	Value
-template	Clinical Trials Phase I
-from_attach	ALL
-to_attach	TARGET
-exclude_type	ItemRevision

- The following argument settings launch multiple subworkflows based on the **Clinical Trials Phase I** workflow template. Each item revision that was a target or reference attachment of the parent workflow launches a new subworkflow with that item revision as the target.
  For example, if the parent workflow contained three item revisions as targets, three different subworkflows are launched.

Argument	Value
-template	Clinical Trials Phase I
-from_attach	ALL
-to_attach	TARGET
-include_type	ItemRevision
-multiple_processes	

3. Creating task templates

Create a task template

A *task template* is a task that you can apply and reuse in workflow templates. These are based on a pre-existing Workflow Designer task. You can then edit that task to offer new capabilities.

The **Task** template is the default template of the EPMTaskTemplate type. Use it as a starting point for creating your own tasks, such as a review task to include a specific set of key roles, or other site-specific tasks for users to complete.

1. In the Workflow Designer, click **New** > **Create Workflow Template**.
2. Click **Task** as the **Template Type**.
3. Enter a unique **Name** and **Description**.
4. Under **Based On**, select an existing template that the new task must be based on.

   The **Type** list automatically populates with the task type for your new template.
5. Click **Add**.

   The new task is added to the **Templates** panel. You can open it and edit the task properties.

   The new task is also available in the **Task Palette** and can be included in workflow templates.

Create custom task templates

What are custom task templates?

You can create your own specific task requirements with a *custom task template*. This is a custom-designed task that you can apply and reuse in workflow templates. It is based on a pre-existing task that you can then edit to create your own.

For example, you may want a task to perform an external third-party authorization. Such a custom task type can be created in BMIDE and then associated with a custom task template in Active Workspace. Modifications to the task can be made using the **Handlers** tab, such as adding a custom authentication handler. The new custom task template can then be made available.

Configuration requirements

Creating a custom task template requires BMIDE configuration before you can start the creation process in the Workflow Designer. Additional configuration is also required to designate an associated icon to

the template as well as for configuring the task behavior and adding the custom task template to the Task Palette.

Create new business types for custom task templates

Administrators must add two new business types for the custom task template in BMIDE. Create one for **EPMTaskTemplate** and one for **EPMTask** to be referenced at runtime.

The two new types are:

- **EPM<YOUR_CUSTOM_TASK>TaskTemplate**
  Include a subtype of **EPMTaskTemplate**.

- **EPM<YOUR_CUSTOM_TASK>Task**
  Include a subtype of **EPMTask**.

Create a custom task template

1. In the Workflow Designer, click **New** > **Create Workflow Template**.

2. Click **Task** as the **Template Type**.

3. Enter a unique **Name** and **Description**.

4. Under **Type**, select the name of the custom type added in BMIDE.

5. Click **Add**.

   The newly created task template is added to the list of templates in the **Task Palette**.

   Additional configuration is needed to create the task icon and to customize the task behavior in the Workflow Designer.

Configure a custom task template

Once you create the new custom task template in Workflow Designer, you can add an icon and customize its behavior.

Add an icon to a task

Custom icons are a good way to offer better visual representation about where a certain task is in a workflow or what its custom function is. To designate an icon for the new template, see Adding custom type icons in the Active Workspace documentation.

Configure the task behavior

Once the icon has been associated to the new task template, you can specify the task behavior in the Workflow Designer.

Configure Perform UI for the custom task

Active Workspace contains the file **performTaskPanelConfiguration.json** that is instrumental in building your task behavior. This file provides the binding for the out-of-the-box **Perform Task** panels that are displayed in the secondary area binding for the secondary area (in the **Overview** tab) or in the right-side **Tool and Information** command area.

The out-of-the-box configuration JSON file is as follows:

```
{
    "secondaryAreaContribution":
    {
      "Awp0EPMConditionTaskPerform" : {
            "id": "Awp0EPMConditionTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMConditionTaskPerform",
            "condition": "modelObject.type === 'EPMConditionTask'"
        },
        "Awp0EPMDoTaskPerform" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMTaskPerform",
            "condition": "modelObject.type === 'EPMDoTask'"
        },
        "Awp0EPMSSTTaskPerformSecondary" : {
            "id": "Awp0EPMSSTTaskPerformSecondary",
            "priority": 1,
            "performPanelId": "Awp0EPMSSTTaskPerformSecondary",
            "condition": "modelObject.type === 'EPMSelectSignoffTask'"
        },
        "Awp0EPMRouteTaskPerformSecondary" : {
            "id": "Awp0EPMRouteTaskPerformSecondary",
            "priority": 1,
            "performPanelId": "Awp0EPMRouteTaskPerformSecondary",
            "condition": "modelObject.type === 'EPMRouteTask'"
        },
        "Awp0SignoffPerform" : {
            "id": "Awp0SignoffPerform",
            "priority": 1,
            "performPanelId": "Awp0SignoffPerform",
            "condition": "modelObject.type === 'EPMPerformSignoffTask' ||
             modelObject.type === 'Signoff'"
        },
        "Awp0EPMTaskPerform" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMTaskPerform",
            "condition": "( (modelObject.type != 'EPMConditionTask' && modelObject.type !
= 'EPMSelectSignoffTask' && modelObject.type != 'EPMRouteTask' && modelObject.type !=
'EPMDoTask' && modelObject.type != 'Signoff' &&  modelObject.type !=
'EPMPerformSignoffTask') && modelObject.modelType.typeHierarchyArray.indexOf('EPMTask')
```

```
>= 0) "
        }
    },
    "perfromTaskToolAreaContribution":
    {
       "Awp0EPMConditionTaskPerform" : {
            "id": "Awp0EPMConditionTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMConditionTaskPerform",
            "condition": "modelObject.type === 'EPMConditionTask'"
        },
        "Awp0EPMDoTaskPerform" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMTaskPerform",
            "condition": "modelObject.type === 'EPMDoTask'"
        },
        "Awp0EPMSelectSignoffTaskPerform" : {
            "id": "Awp0EPMSelectSignoffTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMSelectSignoffTaskPerform",
            "condition": "modelObject.type === 'EPMSelectSignoffTask'"
        },
        "Awp0EPMRouteTaskPerform" : {
            "id": "Awp0EPMRouteTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMRouteTaskPerform",
            "condition": "modelObject.type === 'EPMRouteTask'"
        },
        "Awp0SignoffPerform" : {
            "id": "Awp0SignoffPerform",
            "priority": 1,
            "performPanelId": "Awp0SignoffPerform",
            "condition": "modelObject.type === 'EPMPerformSignoffTask' ||
modelObject.type === 'Signoff'"
        },
        "Awp0EPMTaskPerform" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 1,
            "performPanelId": "Awp0EPMTaskPerform",
            "condition": "( (modelObject.type != 'EPMConditionTask' && modelObject.type !
= 'EPMSelectSignoffTask' && modelObject.type != 'EPMRouteTask' && modelObject.type !=
'EPMDoTask' && modelObject.type != 'Signoff' &&  modelObject.type !=
'EPMPerformSignoffTask') && modelObject.modelType.typeHierarchyArray.indexOf('EPMTask')
>= 0) "
        }
    }
}
```

The correct panel ID is chosen based on the object type as defined in the **Condition** expression of each panel ID. The selected panel ID is set in context, and the respective view is loaded.

Tool and Information area Perform Task Panel View file

The configuration of the **Perform Task Panel View** file (Awp0PerformTaskPanelView.html, in the Tool and Information area) is as follows:

```
<aw-perform-command-panel caption="i18n.performTask">
    <aw-perform-panel-include name="{{ctx.performTaskCtx.activePerformTaskPanelId}}"
exist-when="ctx.performTaskCtx.activePerformTaskPanelId"></aw-perform-panel-include>
</aw-perform-command-panel>
```

Secondary area Perform Task Panel View file

The configuration of the **Perform Task Panel View** file (Awp0PerformTaskView.html, in the secondary area) is as follows:

```
<aw-panel id="Awp0PerformTask" exist-when="data.isTaskPerformable">
    <aw-perform-panel-include name="{{ctx.performTaskCtx.activePerformTaskPanelId}}"
exist-when="ctx.performTaskCtx.activePerformTaskPanelId"></aw-perform-panel-include>
</aw-panel>
```

User customization

Adding a custom perform UI in Active Workspace requires:

- Creation of a custom Active Workspace module.
- Addition of the file **performTaskPanelConfiguration.json** to the **viewModel** folder of the custom module

Use case 1: Include a specific panel for a custom type in a custom task

You can include a specific panel for a custom type in a custom task. For this, you must include the **performTaskPanelConfiguration.json** file in your custom module.

Example of a custom **performTaskPanelConfiguration.json** file:

```
{
    "secondaryAreaContribution":
    {
       "Awp0Fnd2CustomTaskPerform" : {
            "id": "Awp0Fnd2CustomTaskPerform",
            "priority": 2,
            "performPanelId": "Awp0Fnd2CustomTaskPerform",
            "condition": "modelObject.type === 'Fnd2CustomTask'"
        }
    },
    "perfromTaskToolAreaContribution":
    {
        "Awp0Fnd2CustomTaskPerform" : {
            "id": "Awp0Fnd2CustomTaskPerform",
            "priority": 2,
            "performPanelId": "Awp0Fnd2CustomTaskPerform",
            "condition": "modelObject.type === 'Fnd2CustomTask'"
        }
    }
}
```

In this example, the **Awp0Fnd2CustomTaskPerform** panel is added for the custom type **Fnd2CustomTask**. The custom panel ID can be any value, and the corresponding panel is loaded if the condition matches the type of the custom task (**Fnd2CustomTask** in this example). If no custom panel ID is present, then Active Workspace displays the default view as the generic EPM TaskPerform panel.

Use case 2: Display your custom panel for an object

This example explains how to display your own custom panel for the **EPMDoTask** object. In this case, you may require an additional field to display information about the task. You can create this customization by passing the higher priority in your custom .json configuration file as shown here:

```
{
    "secondaryAreaContribution":
    {
        "Awp0EPMDoTaskPerform1" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 2,
            "performPanelId": "Awp0EPMConditionTaskPerform",
            "condition": "modelObject.type === 'EPMDoTask'"
        }
    },
    "perfromTaskToolAreaContribution":
    {
        "Awp0EPMDoTaskPerform1" : {
            "id": "Awp0EPMTaskPerform",
            "priority": 2,
            "performPanelId": "Awp0EPMConditionTaskPerform",
            "condition": "modelObject.type === 'EPMDoTask'"
        }
    }
}
```

The task type provided with Active Workspace has a default priority of 1, but greater integer values have a higher priority. You can override this by providing a higher priority to handle that case. In this example, **Awp0EPMConditionTaskPerform** is displayed for a Do task where the Active Workspace-registered type is **Awp0EPMTaskPerform**.

Use the custom task created in Active Workspace in Teamcenter Rich Client

If you create a custom task in Active Workspace and plan on using that task in the Teamcenter rich client, additional steps are required. See "Creating custom templates" in the Teamcenter help on https://support.sw.siemens.com for instructions.

4. Editing workflow templates

Offline versus online

Deciding whether to edit a workflow template online or offline is determined by whether you want to grant users access to the existing version of the workflow template while you edit it.

- Offline editing prevents users from accessing the workflow template while you edit it. Use this option when you do not want the old version of the template available until your edits are complete.
  With this option, there is only one instance of the template. The system sets the workflow template to the **Availability: Offline** stage. The template is not available to users initiating workflow processes against objects; it does not appear in the **Process Template** list.

- Online editing allows users to initiate workflows based on the existing version of the workflow template while you edit a copy of the same template.
  The system makes a copy of the workflow template and sets it to the **Availability: Offline** stage; this is the version you edit. Both versions of the workflow template appear in the **Process Template** list. The **Availability: Offline** symbol appears next to the version being edited.
  Users can continue to use the existing version of the workflow template. When you switch the edited version to the **Available** stage, the existing copy is overwritten; only the edited copy remains available.

  > Note:
  >
  > Users who do not have the appropriate rights or role will only see **Online** templates.

Edit workflow templates

Edit a template

Using the **Edit** mode in Workflow Designer, you can modify existing templates or templates you are actively developing.

1. Select a workflow template from the list and click **Edit** > **Start Edit**.

   Offline templates are indicated by the **Availability: Offline** icon .

2. The template is offline and ready for editing. In the **Properties** tab, you can change the template name, add or modify instructions, apply filter conditions, and other parameters.

> Note:
>
> You cannot rename a template with an identical name already in use. If a name is in use, an error message appears.

3. When you finish editing, click **Edit** > **End Edits**. This saves your changes, but keeps the template offline.

   Click **Edit** > **Online** to make the template available.

4. If the Workflow Designer is configured to accommodate applying template changes to active processes, additional prompts display; see **Apply Templates to Active Processes** for more information.

Apply template edits to active processes

After editing a workflow template, you can make the template available to users and apply the edits to active processes based on the template. This action modifies all of the template's active processes simultaneously.

The setting configured in the **EPM_enable_apply_template_changes** option determines how and when the processes are applied. Only the **OPTIONAL** or **AUTOMATIC** values apply edits to active processes.

Once you click **Edit** > **Online** to make the template available, you are prompted based on your preferences.

NONE

Any changes are automatically applied to the template but not to any active workflows. This is the default value.

OPTIONAL

Applies workflow template edits to active workflows based on each selected workflow template.

This setting lets you choose on a case-by-case basis whether to apply workflow template edits to active workflows based on the workflow template.

Yes:

At the prompt, if you click **Yes**, your edits are applied to all active workflows based on that workflow template. The request is processed in the background if Dispatcher is configured. Otherwise, changes are applied to the foreground. A confirmation message provides details on the number of workflows that were successfully updated and the number of workflows that failed.

No:

If you click **No** at the prompt, the changes are not applied to active workflows.

AUTOMATIC

All changes are automatically applied to active workflows.

If Dispatcher is configured, the template changes are automatically applied in the background, without any prompting.

If Dispatcher is not configured, the template changes are applied directly to the foreground and a confirmation message displays how many workflows (if any) include the changes to the template.

> Note:
>
> Dispatcher must be enabled and configured for asynchronous processing.

Edit the template layout

There are multiple layout options for a template.

The preference **WRKFLW_preferred_diagram_layout** controls the preferred layout. This setting determines which layout appears when you first display a template. It applies this layout to both the Workflow Designer and Workflow Viewer.

To edit the layout of a workflow template, do the following:

1. Open the template that you want to modify, and click **Edit** > **Start Edit**.
2. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.
3. Click **Apply Layout** and select a layout option.

   Auto Layout includes:

   - Top-to-Bottom
   - Right-to-Left
   - Left-to-Right

   Fixed Layout

   This layout option displays the task (and subtask) node size and location as they were last saved in the Teamcenter rich client or Active Workspace. This also lets you view legacy template layouts as

they were originally designed in the Teamcenter rich client. While editing a template, you can manually set the location of the template nodes. Fixed Layout node positions are automatically saved as they are added or moved.

Switching from Auto Layout to Fixed Layout

When switching from Auto Layout to Fixed Layout, the previously saved task node locations are displayed. For templates that have just been created in Active Workspace, the nodes in the Fixed Layout are initially the same as what they were in Auto Layout.

Switching from Fixed Layout to Auto Layout

When switching from Fixed Layout to Auto Layout, the task node locations generated by the automatic positioning display. The layout changes made to the nodes are not saved unless you click **Save Layout**. Doing this overwrites all pre-existing location data for the template.

Add filter conditions to templates

Filter conditions set on a template control who can see and use specific templates. Administrators define the conditions and set their parameters. Siemens Digital Industries Software recommends that you use Business Modeler IDE conditions to associate templates with a target object type and a user group. Conditions offer greater versatility, with criteria such as session group, role, and user; target project and target release status; and custom criteria, both session-specific and target-specific, that an administrator can create. See *Use conditions to filter workflow template availability* in the Configure your business data model in BMIDE guide.

To add filter conditions to a template:

1. Select a template and click **Edit** > **Start Edit**.

2. To access the **Filter Conditions**, either select the template background or the **Start** task.

   > Note:
   >
   > Filter conditions are not displayed in the **Properties** tab if a task is selected.

3. Select one of the predefined conditions from the **Filter Conditions** list.

4. Click **End Edit** when all your template changes are complete.

Filtering conditions

The preference **CR_allow_alternate_procedures** manages the visibility of workflow templates in the **Submit to Workflow** panel in Active Workspace. Values include:

- **Any**: The **All** option is pre-selected and all templates are listed when a user submits a workflow.

- **Assigned**: This is the default value. The **Assigned** option is pre-selected when a user submits a workflow. Only templates that match the condition criteria (as defined in the BMIDE condition) are displayed.

- **None**: Only valid templates that match the condition criteria (as specified in the BMIDE condition) are displayed. The options **All** and **Assigned** are not visible when a user submits a workflow.

For example, define the following expression to filter workflow templates while initiating a workflow for a problem report.

```
((0 !=null) AND
u.fnd0ConditionHelper.fnd0isSubTypeOf(o,"ProblemReportRevision")) OR ((t!
=null) AND u.fnd0ConditionHelper.fnd0isSubTypeOf
(t,"ProblemReportRevision
```

The condition expression validates when the object submitted is not null and the object type is **ProblemReportRevision**.

Adding targets after submitting a workflow

It may be necessary to add additional targets after a workflow has started. This is still governed by ACL access on the task first. If you have permission then you can add additional targets.

- If the **CR_allow_alternate_procedures** preference is set to **any** or **Assigned**, there is no restriction on what type of target can be added.

- If the **CR_allow_alternate_procedures** preference is set to none and WRKFLW_allow_adding_target_behavior is set to **1** (default value),there is no restriction on what type of target can be added.

- If the **CR_allow_alternate_procedures** preference is set to **none** and **WRKFLW_allow_adding_target_behavior** (explained as follows) is set to **0**, targets can be added that satisfy the condition criteria of the template.

To allow targets to be manually added or pasted at a specific task level, use the preference **WRKFLW_allow_adding_target_behavior**. This is useful when adding a target after a workflow is initiated. Values include:

- **0**: Allows specific target(s) that satisfy the **fnd0FilterCondition** condition when the value of **CR_allow_alternate_procedures** preference is set to **none**.

- **1**: Allows any target(s). This is the default value.

Configure a default workflow template for an object

You can create a default template for a select set of object types in Workflow Designer. This template is then automatically selected when a user submits an existing object to a new workflow.

Workflow Designer automatically creates a preference, **WorkspaceObject_default_workflow_template**, that associates the template with that class. Note the following:

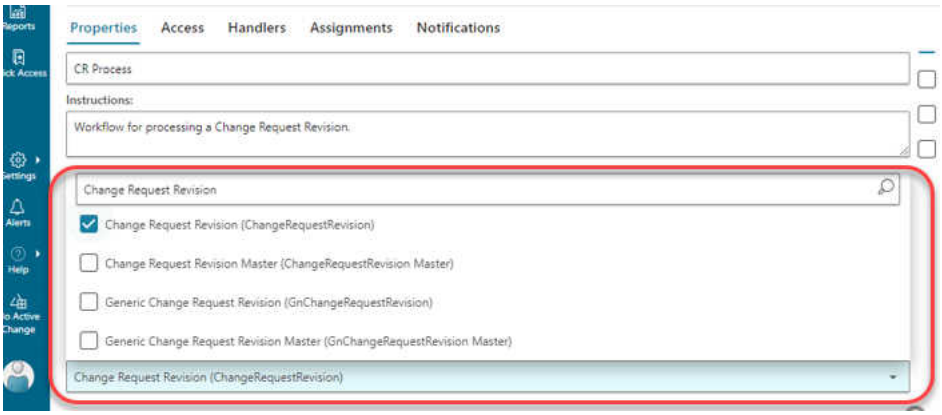
- You can associate multiple classes with a template. A preference is created for each class type.
- If an existing preference already exists and you choose that same class on a new template, the preference is updated with the new template name.
- If you manually add or update a preference, the template reflects the change in Workflow Designer.
- If you delete a preference that the template was associated with, the preference no longer shows the association in Workflow Designer.

To associate a default template for an object:

1. Select a template and click **Edit** > **Start Edit**.
2. To access the default types for the template, either select the template background or the **Start** task.

   > Note:
   >
   > Types are not displayed in the **Properties** tab if a task is selected.

3. Click the **Default for Types** list and select a class to associate with the default template.



4. Click **End Edit** when all your template changes are complete.

5. Managing templates in Workflow Designer

Importing and exporting templates in Workflow Designer

Import workflow templates

You can import workflow templates into the Teamcenter database from an exported workflow template file. Importing templates is useful for transferring workflow templates between different Teamcenter sites. The templates must first be exported from a Teamcenter database into an export file, after which you can import the file into the Teamcenter database at another site.

1. Choose **New** > **Import**.

   The system displays the **Import** panel.

2. Select **Choose File** and navigate to the directory containing the exported template.

3. (Optional) If you want the system to continue the transfer if one or more workflow templates fail to transfer, select the **Continue On Error** check box. If one or more workflow templates fail to transfer, the system records transfer errors in its log files, bypasses the failed workflow templates, and transfers the remaining workflow templates.

   If you do not select this option, the system stops the transfer process if one workflow template fails to transfer and only includes in the transfer those workflow templates that transferred successfully.

4. (Optional) If you want the system to overwrite any workflow template of the same name that already exists in the database, select the **Overwrite Duplicate Templates** check box. The system does not display or log any errors.

   Select this option when the imported workflow template contains changes that you want applied to the database.

   For example, you have added two custom tasks to a workflow template and thoroughly tested the revised template in your test database. Now you are ready to import the changes to the production database. By choosing to overwrite duplicate templates when importing the workflow template to the production database, you are effectively editing the workflow template. On import, the original workflow template is overwritten by the importing workflow template; it now contains the two custom tasks.

   If you do not select this option, any importing template with the same name as an existing template is ignored and the import process continues. A message is logged that a workflow template of the same name exists.

5. (Optional) If you chose to overwrite duplicate templates, you can also choose ignore the origin ID of the template you are importing by selecting the **Ignore origin ID** check box.

   Select this option if you get the following error when attempting to import workflow templates:

   ```
   The importing template(s) do not match with the existing template(s).
   The import of template(s) in overwrite mode failed.
   ```

6. Click **Import** to import the templates contained within the file you selected into the Teamcenter database.

   The imported template names now exist in the database and appear in the template list.

Export workflow templates

You can export workflow templates from the Teamcenter database in XML format, storing the templates in a single export file. After exporting the templates, you can import the file into the Teamcenter database at another site. You can also search the XML to determine handler and argument usage.

1. Select one or more templates to export.

2. Choose **Share** > **Export**.

   The **Export Workflow Templates** panel appears.

3. Specify the name of the export file in the **File Name** box, for example, **template_export**.

   > Note:
   >
   > The export file location is your default downloads folder of your browser.

4. (Optional) If you want the system to continue the transfer if one or more templates fail to transfer, select **Continue On Error**. If one or more templates fail to transfer, the system records transfer errors in its log files, bypasses the failed templates, and transfers the remaining templates.

   If you do not choose this option, the system stops the transfer process if one template fails to transfer and only includes in the transfer those templates that transferred successfully.

5. Click **Export** to export the templates.

   The selected templates are exported in XML format to the file name you defined in step 3.

Note:

A template that is offline and under construction can be exported and then imported into another Teamcenter environment. The offline state is retained during the import process.

Select and compare workflow templates

You can view, select, and compare templates from the Workflow Designer dashboard.

1. Select any template in the list to display the workflow structure along with the workflow **Properties**.

2. Select multiple templates by holding down the **Control** key to select specific templates, or use the **Shift** key to select a range of templates.

   The number of templates selected is shown under **Selected Summary**.

3. Select **Compare** to show all of the selected templates and their details.

   You can compare them for specifics such as Owner or Date Modified.

4. Select **Settings** to customize the workflow criteria available for comparison.

5. To remove the selections, either deselect each template, or double-click **Selection Mode**.

Delete templates

If a template is no longer used or becomes obsolete you can permanently delete it from the system.

Caution:

Deleting templates is permanent and cannot be reversed.

1. Select one or more templates to remove from the system.

2. Choose **Edit** > **Delete**.

3. Choose **Delete** from the dialog box to permanently remove the template.

6. Managing Task Behavior

Delete a task

1. Open the template containing the task you wish to delete, and click **Edit** > **Start Edit**.
2. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.
3. Click the task that you want to delete.
4. Click **Remove**.
   The selected task and any attached links are deleted.
5. Click **End Edit** to save your template.

Modifying tasks

Modify tasks in Workflow Designer

Modify task behavior within a workflow process template using attributes, access control lists (ACLs), handlers, notifications, and properties.

1. Select a template from the list.
2. Click **Edit** > **Start Edit**.
3. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.
4. Select the relevant task in the workflow.

   Modifications can be made to the tasks through the **Properties**, **Access**, **Handlers**, **Assignments**, **Notifications**, and **Forms** tabs.

Specify an ACL for a task

Objects can use an access control list (ACL) to create exceptions to rules-based protection.They define access to an object and determine what operations can be performed on it. Access Manager then applies access as defined by the ACL. See *Access control lists* in Getting Started with Access Manager for more information.

Working with Access Manager

The **EPM-set-rule-based-protection** handler indicates that an ACL will be passed to Access Manager. Access Manager will then apply access as defined by the ACL. Although **EPM-set-rule-based-protection** will indicate an ACL, it does not apply the ACL. Access Manager picks up the ACL and applies and enforces it. The ACL set by **EPM-set-rule-based-protection** is not exposed to Access Manager until a task's state has been set to Started. A task's state does not transition to Started until all the handlers on the Start action execute successfully. This means that other handlers that are located on the same Start action as **EPM-set-rule-based-protection** will not execute under the access indicated by **EPM-set-rule-based-protection**. In order for other handlers to adhere to the access indicated by **EPM-set-rule-based-protection**, they can either be placed on the Complete action of the current task or the Start action of a successor task. It is important to understand this concept because some handlers rely on access, and therefore a proper configuration is required to ensure the intended access is being applied when these handler execute.

For example, a desired configuration may be to have all of the following happen on a single task:

1. **EPM-set-rule-based-protection** indicates an ACL.

2. Access Manager picks up the ACL and applies the access.

3. **EPM-attach-related-objects** executes based on the access of the ACL that was indicated by **EPM-set-rule-based-protection**.

Since the ACL will not be applied until the task starts, to achieve the desired behavior, the **EPM-attach-related-objects** should be placed on the Complete action.

In this example, the configuration should look like this:

- Task Start handler: EPM-set-rule-based-protection

- Task Complete handler: EPM-attach-related-objects

With this configuration the processing will execute as follows:

1. Current task state is Pending.

2. Task is triggered to start.

3. Handlers on the Start action will execute, which in this example is the **EPM-set-rule-based-protection** handler, and **EPM-set-rule-based-protection** will indicate an ACL.

4. After the handlers on the Start action execute successfully, the task state is set to Started. The indicated ACL will now be applied (and this access will remain until a different access is set).

5. Handlers on the Complete action will execute, which in this example is the **EPM-attach-related-objects** handler.

6. The **EPM-attach-related-objects** executes under the intended ACL, which is the ACL indicated by **EPM-set-rule-based-protection** in step 3.

Specify an ACL in Workflow Designer

In Workflow Designer, use the **Access** tab to configure a task with a specific ACL for Access Manager-enabled systems.

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.

   - Keep the template offline during the editing process.

3. Click the task to modify.

4. Click the **Access** tab.

5. Click the type of ACL to apply:

   - **System**
     These ACLs can be applied to the task and all subsequent tasks in the workflow process unless it is changed by another instance of the **EPM-set-rule-based-protection** handler or the process is completed.

   - **Workflow**
     These ACLs can stay in effect for the task until the same workflow sets another ACL later in the process or the process is completed.

6. Click the list to select a specific ACL.

7. Click **End Edit** to save the changes to the workflow.

The Workflow Designer automatically creates and updates the **EPM-set-rule-based-protection** handler once the edits are saved.

Note:

If you create or update the **EPM-set-rule-based-protection** handler from **Handlers** tab, the **Access** tab automatically reflects those changes.

Work with task handlers

Handlers are the lowest-level building blocks in a workflow. They are small ITK programs used to extend and customize tasks. There are two kinds of handlers:

- Action handlers extend and customize task actions. They perform such actions as displaying information; retrieving the results of previous tasks (inherit); notifying users; assigning users, reviewers, dynamic participants; setting object properties and protections; and adding release status and trigger system commands.

- Rule handlers integrate workflow business rules into workflow processes at the task level. They attach conditions to a task action. Rule handlers confirm that a defined rule has been satisfied. If the rule is met, the handler returns the **EPM_go** command, allowing the task to continue. If the rule is not met, it returns the **EPM_nogo** command, preventing the task from continuing. If there are multiple targets for a single rule handler, all targets must satisfy the rule for **EPM_go** to be returned (**AND** condition).
  Rule Quorums
  Many conditions defined by a rule handler are binary (that is, they are either true or false). However, some conditions are neither true nor false. EPM allows two or more rule handlers to be combined using logical **AND/OR** conditions. When several rule handlers are combined using a logical **Or** condition, rule quorums specify the number of rule handlers that must return **EPM_go** for the action to complete.

Action and rule handlers in the **Handlers** panel can be copied:

- From one action to another action in a task.

- From one task to another task in the same template.

- From a task in one template to a task in another template.

A full list of action and rule handlers and their definitions is available in the Workflow Handlers Guide.

Adding task handlers

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.

- Keep the template offline during the editing process.

3. Click the task to modify.
4. Click the **Handlers** tab.
5. Click **Add** ⊕ for a rule or an action handler.
6. In the **Add** dialog box, select a **Trigger** from the list.
7. Select a **Handler** from the list.

   The Handler list is not case-sensitive. All the applicable handlers are available in the list. This list may contain custom handlers as well as handlers that are included with other solutions.

   Click **Help** ⓘ to launch the handler documentation.
8. Add a new argument row by clicking **Add** ⊕ under **Arguments**.
9. In the **Add Arguments** dialog box, enter the new argument name in the **Arguments** field.
10. Enter the corresponding values in the **Value(s)** field.

    Hints are provided for the handler arguments and their respective values to help you identify mandatory and optional parameters and avoid runtime errors.
11. Click **Add**.

    The new argument is listed in the **Arguments** table. To make any changes to the argument, highlight the row and click **Start Edit** ✎. Click **Remove** ⊖ to delete an argument.
12. Click **Add** when the handler information is complete.

    The new handler is displayed in the rule or action handler table. To show all the information in a cell, click the gear icon ⚙ to **Wrap Text**.

Removing task handlers

1. Open the template that contains the tasks you want to modify, then click **Edit** ✎ > **Start Edit**.
2. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the task to modify.

4. Click the **Handlers** tab.

5. In the table, highlight the action or rule handler that you want to delete and click **Delete** ⊖.

Updating task handlers

1. Open the template that contains the tasks you want to modify, then click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the task to modify.

4. Click the **Handlers** tab.

5. In the table, highlight the action or rule handler that you want to edit.

   The available editing options include:

   Edit

   Click this option to display the **Edit** dialog box where you can edit the predefined handler. You can select a different **Trigger** from the list.

   Edit existing arguments in the **Arguments** table by highlighting the row and clicking **Start Edit**. You can enter a different value for the argument in the **Value(s)** field, and click **Save**.

   To remove an argument, highlight it and click **Delete**.

   Click **Save** when edits to the handler are complete.

   Cut

   Removes the handler from the current location and saves it to the clipboard so you can paste it elsewhere.

   Copy

   Creates another instance of the same handler in a new location, so you have two different instances of the same handler.

Paste

Places the copied handler in the selected location.

Move Up

Moves the selected handler up within the table. This icon only appears for handlers within the same trigger group.

Move Down

Moves the selected handler down within the table. This icon only appears for handlers within the same trigger group.

Delete

Removes the handler from the table.

Viewing task handlers

You can view the task handlers that make up each task in a workflow.

1. Open the template that contains the tasks you want to view.
2. Click the task that contains the handlers you want to view.

   To view handler information for the root task, click the initial **Start** task. 
3. Click **Handlers** to view the **Rule Handlers** and **Action Handlers** assigned to the task.

   Click **Information** to view handler details.

Assign users to tasks

Use the **Assignments** tab to assign a user to a task.

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.
2. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the task to assign a user to it.

4. Click the **Assignments** tab.

5. Click **Add** ⊕ under **Assignments** to view a list of assignees.

6. Click **Add** ⊕ next to the **Assignee** type.

7. Add users from the list by **Key Roles**, **Teams**, or (as individual) **Users**.

   Key Roles

   Specifies the key roles defined for workflows, dynamic participants, and project management.

   Teams (Resource Pools)

   Specifies predefined teams to choose from.

   Users

   Specifies participants in a specific group or role.

8. Select an assignee, and click **Add**.

9. Click **Add** to save.

   The new task assignment is listed in the table. To make any changes, highlight the assignment and click **Edit** ✎.

Work with Review tasks

Review tasks route workflow targets (documents, parts, designs, and so on) for review and require additional information.

The Review task includes two subtasks:

- The **select-signoff-team** subtask requires the workflow process initiator to select the users performing the review (the signoff team). You can configure this subtask with predefined group or role profiles. The workflow process initiator must select or allow the workflow process initiator to select users of their choice in an ad hoc manner. Each member of the signoff team is responsible for reviewing and signing off on the attached target objects.
  This subtask allows the selector to search by group, role, user and to select signoff members individually or by project teams.

- The **perform-signoffs** subtask is then distributed to the selected signoff team, prompting them to review the target objects and sign off. When a Review task is performed in a workflow process, the

perform-signoffs task displays these options to each signoff team member: **Approve**, **Reject**. Selecting either **Approve** or **Reject** performs the task.

> Caution:
>
> Do not add or delete subtasks from the **Review** task. It may cause an error that prevents the task from being executed.

To modify a Review task, do the following.

1. Open the template that contains the Review task you want to modify, then click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.

   - Keep the template offline during the editing process.

3. Click the Review task that you want to modify.

Configure an approval quorum

You can specify an approval quorum for the task. An *approval quorum* is the minimum number of signoff members required for a Review task to proceed.

1. Once you select the Review task to add the quorum to, click the **Assignments** tab.

2. Under **Participation**, specify the type of approval quorum: **Percent** or **Numeric**.

3. Accordingly, enter one of the following in the text box:

   - A percentage value required to progress the task. For example, if it is set to 51%, then 51% of signoff members must approve for the task to move ahead.

   - A numeric value to specify the number of signoff members needed to move the task ahead. For example, if it is set to 5, then 5 members must approve for the task to move ahead. This value cannot exceed the number of reviewers added.

4. Click **Require full participation** if you want all users to review and comment before the task progresses.

Add assigners and reviewers to the task

1. Once you select the Review task to modify, click the **Assignments** tab.

2. Under **Assignments**, click **Add** ⊕ to view a list of assignees.

3. Click **Add** ⊕ next to **Assigner**.

   Add users from the list by **Key Roles**, **Teams**, or (as individual) **Users**.

   Key Roles

   Specifies the key roles defined for workflows, dynamic participants, and project management.

   Teams

   Specifies predefined teams to choose from.

   Users

   Specifies participants for a specific group or role.

4. Select an assigner, and click **Add**.

5. Click **Add to** ⊕ next to **Reviewers**.

   Add users from the list by **Key Roles**, **Teams**, or (as individual) **Users**.

   Key Roles

   Specifies the key roles defined for workflows, dynamic participants, and project management.

   Allow reassign outside selected group and role lets the task owner reassign the workflow to another group and role outside of what is selected. If you do not select this check box, the task owner can only reassign the workflow to someone in the same group and role.

   Teams (Resource Pools)

   Specifies predefined teams to choose from.

   Enter the **Number of Reviewers** to review the task.

   Under **Assignment Option**, click one of the following:

   - **Team Assignment to be selected by assigner**
     This option adds the team assignment to the select-signoff-team.

   - **Team Assignment to be claimed by assignee**
     This option creates resource pool assignments for the group or role that was specified, such as Engineering or Designer.

Allow reassign outside selected group and role lets the task owner reassign the workflow to another group and role outside of what is selected. If you do not select this check box, the task owner can only reassign the workflow to someone in the same group and role.

Users

Specifies participants for a specific group or role.

Allow reassign outside selected group and role lets the task owner reassign the workflow to another group and role outside of what is selected. If you do not select this check box, the task owner can only reassign the workflow to someone in the same group and role.

6. Once you select your reviewers, click **Add**.

7. Click **All Reviewers must sign off** if you want all your reviewers to sign off on the review without the provision of being removed from the list.

8. Click **Add** to save.

   The new task assignment and its details are listed in the table.

9. Once you select your reviewers, click **Interactively paused for adding more reviewers** to be able to pause the workflow to add more reviewers to the select-signoff-team subtask, if necessary.

   If you do not select this check box, the select-signoff-team task ends automatically with no additional reviewers, and continues to the perform-signoff subtask. The associated handler under the **Handlers** tab shows **-auto_complete**.

Add notifications to tasks

Add notifications to tasks in the **Notifications** tab.

1. Open the template that contains the tasks you want to modify, then click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the task to add the notifications to.

4. Click the **Notifications** tab.

5. To create an email notification for a task, click **Add** ⊕.

6. In the **Notify When Task** list, select when to notify the recipient during the task's lifecycle.

7. Enter a **Subject** for your notification email.

8. Enter the **Message** text for the recipient(s).

9. (Optional) Choose to include any information in the email about **Process Info**, **Targets**, and **References**.

10. Click **Add** ⊕ next to **Recipients** to identify whom to notify.

    Add users from the list by **Key Roles**, **Teams**, or (as individual) **Users**.

    Key Roles

    Specifies the key roles defined for workflows, dynamic participants, and project management.

    Teams (Resource Pools)

    Specifies predefined teams to choose from.

    Users

    Specifies participants for a specific group or role.

11. Select the recipient(s) to be notified, and click **Add**.

12. Click **Add** to save.

    The notification and its associated information is added to the notifications table. To remove a notification, highlight the notification and click **Remove** ⊖. To edit the notification, click **Edit** ✎ to display the Edit dialog box. To show all of the information in a cell, click the gear icon ⚙ to **Wrap Text**.

Work with forms

Add forms to tasks in the **Forms** tab. The **Forms** tab is displayed when you include a **Task** task in your workflow.

1. Open the template that contains the tasks you want to modify, and click **Edit** ✎ > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.

   - Keep the template offline during the editing process.

3. Once you drag a **Task** task to the Workflow Designer from the **Task Palette**, the **Forms** tab is displayed.

4. On the **Forms** tab, click **Add** ⊕ to create a new form for the task.

5. In the **Add** dialog box, enter a **Form Type** or select one from the list.

6. Enter a **Form Name**.

7. Enter a **Description** for the form.

8. Click the **Attach As** list to select whether to add the form as a target or a reference.

9. Click **Add**.

   The new form is added to the **Forms** table and the **EPM-create-form**, **EPM-display-form**, and **EPM-hold** handlers are added to the **Handlers** tab. To make any changes to the form, click **Edit** ✎, or click **Delete** ⊖ to remove the form and its associated handlers from the task.

Modify task properties

You can modify task properties in the **Properties** pane.

1. Open the template that contains the tasks you want to modify, then click **Edit** ✎ > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the task to modify.

4. Click the **Properties** tab.

5. Edit the **Name** if needed.

6. Enter any **Instructions** for the task.

7. The **Type** list identifies the task type.

8. Enter the duration for the task under **Task Duration (Hours)** in hours. An associated task handler (EPM-set-duration) is created in the **Handlers** table.

9. Click **Show Task in Process Stage List** to display the task in the **Process State List** property for the target object. Tasks in the **Process Stage List** are used to determine the ACL for the target objects.

10. Click **Process in Background** to configure the task (and any subtasks) to allow for background processing.

    This option appears if Workflow Designer has been configured for background processing. If so, all of those tasks' actions (except **Perform** and **Assign**) are processed asynchronously.

    Values for **EPM_task_execution_mode** include the following:

FOREGROUND	All tasks run in the foreground. This is the default.
BACKGROUND	All tasks run in the background. This option displays the **Dispatcher Request Priority** list on the **Properties** tab for a task.
CONFIGURABLE	• If the value is **On**, and you select the **Process in Background** check box, the task runs in the background. When selected, the **Dispatcher Request Priority** list is displayed as well. • If the value is **Off**, the task runs in the foreground.

11. If the task is running in the background, the **Dispatcher Request Priority** list is displayed. This list lets you define the execution priority for background tasks and is specifically used with multiple instances of Dispatcher running. Click the list to select the level of priority for the task.

12. Click **Select Participants from Workflow** to select dynamic participants from the workflow instead of target objects. This option is only applicable for the root task.

13. Click **Require Task Confirmation on Complete** to require users to confirm that a selected interactive task is completed in Active Workspace.

    Selecting a root task requires completion confirmation on all child tasks.

Add a list of values to Review and Acknowledge tasks

The **Review** and **Acknowledge** tasks have optional properties for task behavior. Adding a **Decision Label** (created in BMIDE) to a **Review** or **Acknowledge** task provides a list of values (LOV) to the task. LOV examples of decisions include choices such as **Approve** and **Reject**.

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.

2. Select one of the following:

   • Keep the template online during the editing process.

   • Keep the template offline during the editing process.

3. Click the task to modify.

4. Click the **Properties** tab.

5. Select a value from the **Decision Label** list.

   The LOV selection adds decision buttons to the workflow task.

Add a release status to an Add Status task

Status tasks have optional properties for task behavior.

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.

2. Select one of the following:

   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.

3. Click the status task to modify.

4. Click the **Properties** tab.

5. Select a value from the **Release Status** list.

   The values in the list can include the default values or custom statuses defined by your organization.

   Adding a release status adds a release status indicator icon to a workflow task.

Work with Condition Tasks

Use the **Condition Task** template to branch your workflow process according to defined criteria. Because this task template is used to branch workflow process flow, you must always create at least two paths branching off from the task. The paths can be either success paths, failure paths, or a combination of the two.

- Success paths can be either true paths, false paths, or paths with a customized result.
- Failure paths can only be generated from manual **Condition** tasks. They allow an alternate course when a specified task is rejected, when a user determines the path cannot be completed, or when an error occurs.

Tip:

If you use a **Condition** task to branch your workflow process, you can use one or more **Or** tasks later in the workflow process to resolve the paths into a single path.

The system determines which of the branches flowing from a **Condition** task to perform based on the *task result*. The task result is stored in the **Condition** task. The successor tasks have a handler configured with a value that may match the task result. After the task result is set, the successor tasks are examined and any successor tasks containing a value matching the task result are started. Use any of the following methods to set the task results:

- Create a query against the target (automatic only).
- Create a query against the task (automatic only).
- Create a query against subworkflow (also known as subprocesses) (automatic only).
  If there are multiple subworkflows, a query runs on the associated subworkflows and the results are used to branch accordingly. The query is typically configured to look at the root task's **result** attribute for all the subworkflows.
  If there is only one subworkflow and it is configured to set the result on the **Condition** task, no query is needed, and the workflow follows the branch based on the result.

A **Condition** task can be configured to run either automatically or manually. You must determine the configuration that is best suited for the workflow process template you are defining. Typically, if a handler can determine the criteria, it is best to configure the task as automatic.

Task	Description
Automatic **Condition** task	Add an action handler that sets the task's result to true, false, or a customized value. The simplest way to achieve this is to define a condition query or BMIDE condition filter for the task at design time; this automatically inserts the action handler. Alternatively, you can write custom code or use the **EPM-set-task-result-to-property** handler.
Manual **Condition** task	During design, you do not define a query or add an action handler to the task template. Because no query is defined and no action handler is configured to set the task result, when the workflow process is run, the end user must manually indicate a value using an interactive dialog box. The value chosen by the end user is used to set the task result. Custom conditions can also appear as manual condition options and appear as buttons in the **Condition** dialog box.

Caution:

To ensure desired results, condition tasks that run queries in workflows should always have at least one target object when a condition query is run against workflow targets.

- When a condition task runs a condition query against workflow targets, the system searches the database for that query class and filters the results based on the workflow target objects.
- Because handlers can move objects between targets and references in a workflow, the workflow may have objects in the references folder, but no objects in the targets folder. The condition query will not search in the database if the workflow does not have any targets. This will set a false path of the condition task.

To configure a **Condition** task, do the following:

1. Open the template that contains the tasks you want to modify, and click **Edit** > **Start Edit**.
2. Select one of the following:
   - Keep the template online during the editing process.
   - Keep the template offline during the editing process.
3. Select the **Condition** task and view the **Properties**.
4. (Optional) **Instructions** are not required for tasks, but may guide users through the task.
5. Continue to the following sections to add condition queries to the task, or configure a task with a BMIDE condition.

Add condition queries to a Condition task

Condition tasks have optional properties for task behavior. Condition queries control the specific functions of a **Condition** task.

Once applied, conditions check for specific parameters that you set. These conditions work in conjunction with the **Success** path from the **Condition** task to its task or target.

1. In the **Properties** tab of the **Condition** task, click **Query**.
2. Select a query from the **Query** list. Queries are created in Teamcenter.

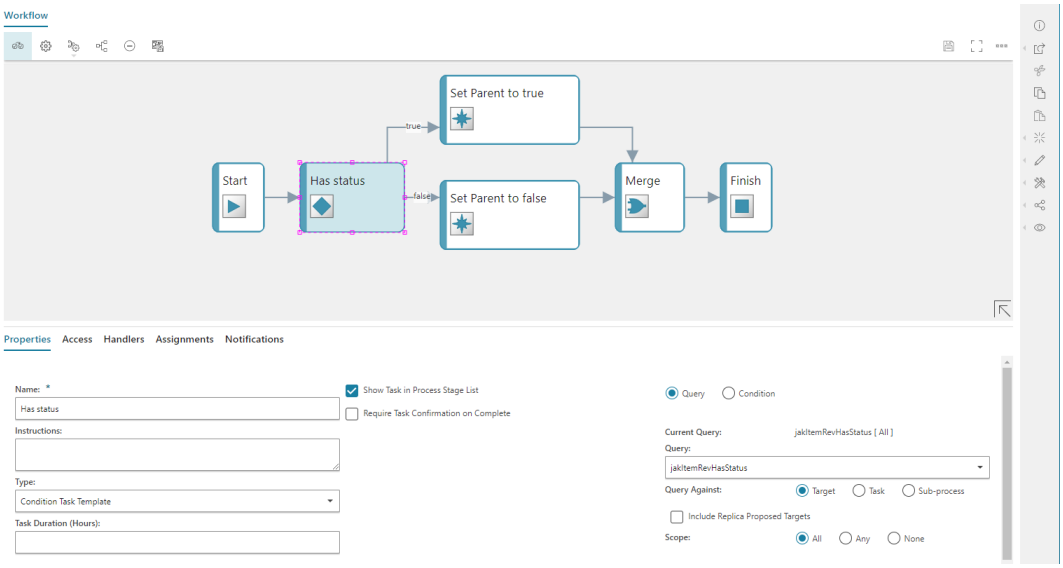
   Your organization may have their own configured queries in this list.

3. Select whether the query should run against a **Target**, **Task**, or a **Sub-process**.

4. When **Target** is selected, **Include Replica Proposed Targets** is active. Select this check box to include targets for the remote workflow task in the search.

5. If a target or subworkflow is selected, apply a **Scope** to the query. Options are **All**, **Any**, or **None**.

   The query is added to the task.



Add a BMIDE condition to a Condition task

You can configure a Configure task with a BMIDE condition to automate the condition task result.

1. In the **Properties** tab of the **Condition** task, click **Condition**.

2. Click the **Condition** drop-down list and select a condition to apply to the task, or enter a search string to locate one.

3. Click the **Include Replica Proposed Targets** check box to include targets for the remote workflow task in the search.

4. Click a scope to determine if the task satisfies the condition for **All** targets, **Any** targets, or **None**.

   The condition is added to the task.