



TEAMCENTER

Workflow Designer on Active Workspace

Teamcenter 2312

Unpublished work. © 2023 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer's applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/base/uca/>, as supplemented by the product specific terms which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/supplements/>.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' Marks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow's idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: support.sw.siemens.com

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

Contents

Overview of Workflow Designer

What is Workflow Designer?	1-1
Before you begin	1-3
What can I do with Workflow Designer?	1-3
Workflow templates	1-4
Access templates	1-4
Workflow elements	1-5
The Workflow Designer interface	1-8

Creating workflow templates

Workflow templates overview	2-1
Basic workflow template process	2-1
Core templates	2-2
Create a workflow template	2-3
Add tasks to a template	2-3
Add paths between tasks	2-4
Delete paths between tasks	2-5
Failure paths	2-5
Task template types	2-7
Customize task paths	2-14
Subworkflows	2-14
What are subworkflows?	2-14
Create a subworkflow	2-16
Creating subworkflows for assemblies	2-18
Creating subworkflows for related objects	2-19

Creating task templates

Create a task template	3-1
Create custom task templates	3-1
What are custom task templates?	3-1
Create new business types for custom task templates	3-2
Create a custom task template	3-2
Configure a custom task template	3-2

Editing workflow templates

Determining which editing options to use	4-1
Offline versus online editing	4-2
Identify templates as non-modifiable	4-3
Edit workflow templates	4-3
Edit a template	4-3
Apply template edits to active processes	4-4
Edit the template layout	4-5

Add filter conditions to templates	4-6
Configure a default workflow template for an object	4-8

Managing templates in Workflow Designer

Importing and exporting templates in Workflow Designer	5-1
Import workflow templates	5-1
Export workflow templates	5-2
Select and compare workflow templates	5-3
Delete templates	5-3
Distributing workflow templates using Multi-Site Collaboration	5-3
Replicate a workflow template	5-3
Synchronize replicated templates	5-4

Managing Task Behavior

Delete a task	6-1
Modifying tasks	6-1
Modify tasks in Workflow Designer	6-1
Access Control Lists	6-2
Work with task handlers	6-9
Assign users to tasks	6-16
Add notifications to tasks	6-23
Work with forms	6-26
Modify task properties	6-26

Using an assignment matrix

What is an assignment matrix?	7-1
The assignment matrix process	7-2
Compare a RASIC table to an assignment matrix	7-3
Administrative tasks	7-5
Create an assignment matrix definition	7-7
Create and populate an assignment matrix	7-8
Update an assignment matrix	7-10
Add and modify the AMX handler	7-11

1. Overview of Workflow Designer

What is Workflow Designer?

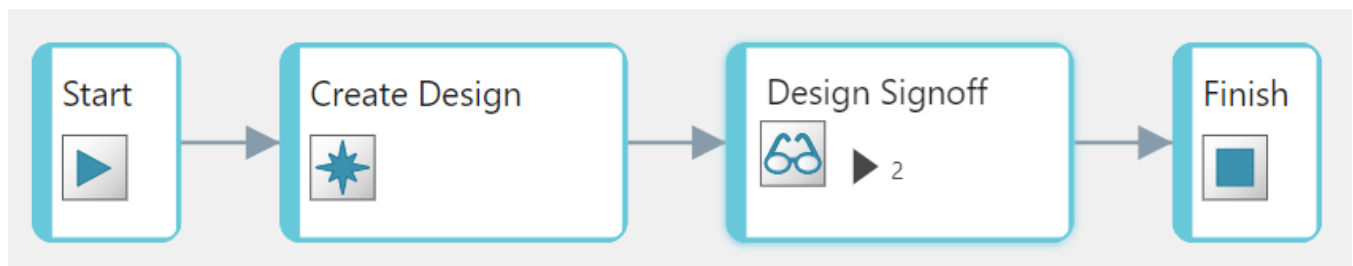
Workflow Designer lets you design workflow templates that can incorporate and automate your company's business practices and procedures. Business users then use these templates to initiate workflow processes.

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.

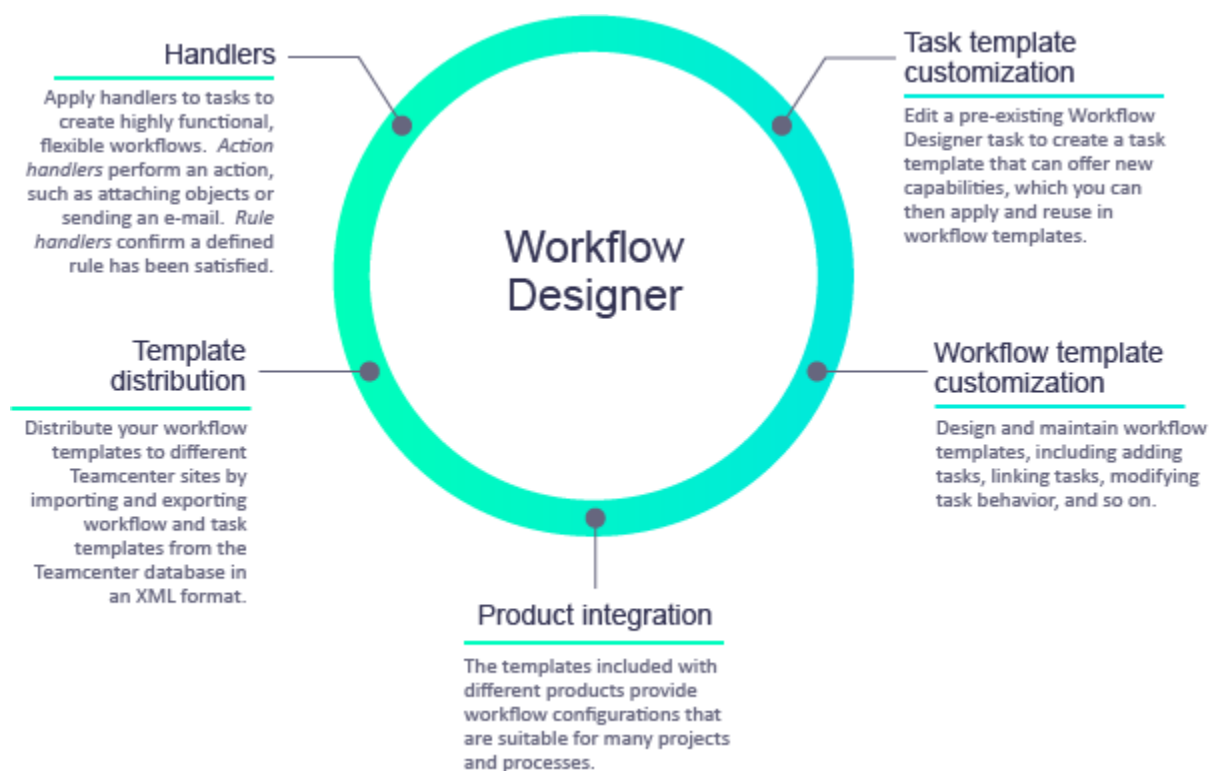
A *workflow* contains a series of tasks to complete in order to reach a repeatable business goal. Customer business objects (such as documents and parts) are passed between participants during a workflow process.

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	<p>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.</p> <p>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.</p> <p>For details, see Access templates.</p>
Get me started	<p>Take a look at the prerequisites and information on how you can enable, configure, and start Workflow Designer before you begin.</p>
Working with templates in Workflow Designer	<ul style="list-style-type: none"> • Create a workflow template • Import workflow templates

	<ul style="list-style-type: none"> • 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
Create privileged users	Allow non-DBA users to create and edit templates
 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.

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 users 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 **Awp0IsPrivilegedWorkflowTemplateAuthor** condition has the same privileges as **Awp0IsWorkflowTemplateAuthor**, but also allows for modifications to templates that are marked as read-only (**Require admin privilege for modification**, found in a workflow's properties).

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-Workflow Designer 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.

Allow non-DBA users to create, edit, import, and export templates

If an administrator wants to include a group of users that are granted Workflow Designer on Rich Client privileges, non-DBA users can be identified to create, import, export, and edit templates.

To identify such users, administrators must do the following:

1. Modify the BMIDE condition **Awp0IsWorkflowTemplateAuthor** to specify a group that includes users with template privileges. This example allows users in the **AuthorGroup** to create and modify templates.

```
Awp0IsWorkflowTemplateAuthor ( UserSession u)
((u.fnd0ConditionHelper.awp0GetProperty(u.group,"privilege","1",false)) OR
((u.fnd0ConditionHelper.awp0IsPlatformVersionAtLeast( "14.3","14.3" ) = true) AND
(Condition::Awp0IsPrivilegedWorkflowTemplateAuthor(u) = true)) OR u.group_name="AuthorGroup"
```

2. Create a new group using the name you specified in the condition, then add users who should be granted template privileges. Or, apply that specified name to an existing group to provide template privileges.

Note:

Non-DBA users can add existing ACLs to a template but cannot create new ones.

Workflow elements

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

Term/Concept	Description
Workflow process	A <i>workflow process</i> 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	<i>Parent workflow processes</i> are workflows that 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	A <i>subprocess</i> is the child workflow of a parent workflow process.

Term/Concept	Description
Workflow template	<i>Workflow templates</i> are 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	<i>Tasks</i> are the fundamental building block used to construct a process. 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 <i>root task</i> . The root task is the top-level parent task that contains all the other tasks as subtasks.
Subtask	A <i>subtask</i> is the child task of a parent task.
Container tasks	<p><i>Container tasks</i> are tasks that contain other tasks:</p> <ul style="list-style-type: none"> • Review <p>Contains select-signoff-team and perform-signoffs tasks. The Decision options are Approve, Reject, and No Decision.</p> • Acknowledge <p>Contains select-signoff-team and perform-signoffs tasks. The Decision options are Acknowledged and Not Acknowledged.</p> • Route <p>Contains Review, Acknowledge and Notify tasks.</p> • Task <p>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.</p>
Interactive tasks	<p><i>Interactive tasks</i> are tasks that require user interaction and display in the affected user's worklists. Different types of tasks have different interactive requirements. Typical tasks include:</p> <ul style="list-style-type: none"> • select-signoff-team <p>The assigned user is required to select a signoff team to sign off the target object of the task.</p>

Term/Concept	Description
	<ul style="list-style-type: none"> • perform-signoffs <p>Assigned users are required to review and sign off the target object of the task.</p> <ul style="list-style-type: none"> • Do <p>The assigned user is required to review and perform the task instructions, then mark the task complete.</p> <ul style="list-style-type: none"> • Route <p>Uses the Review, Acknowledge, and Notify subtasks, each of which has its own dialog box.</p> <ul style="list-style-type: none"> • Task <p>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.</p>
Workflow handlers	<i>Workflow handlers</i> are 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	<i>Task attributes</i> are attributes that further configure task behavior. You can set security attributes, customize task symbols, and define condition results.
Quorum requirements	<i>Quorum requirements</i> are values that specify the number of approvals required before perform-signoffs tasks can complete and workflows can proceed.

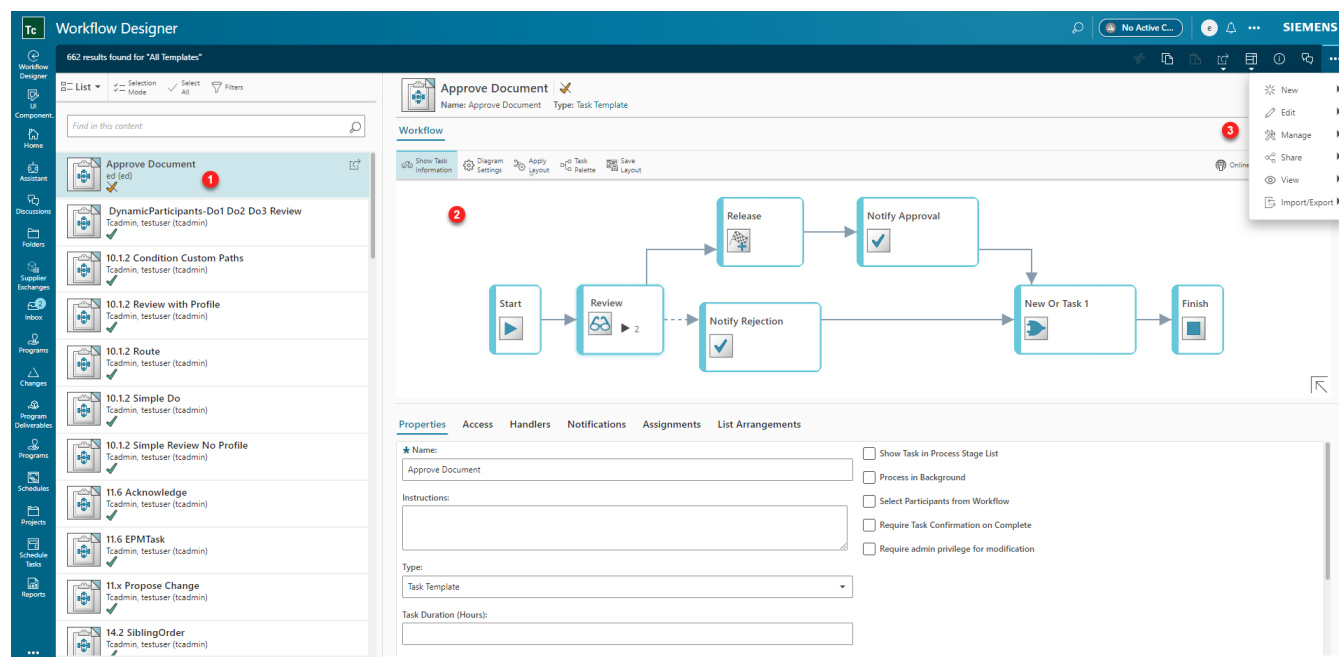
In a workflow, actions are assigned or allowed depending on the type of user.

User	Description
Responsible party	A <i>responsible party</i> 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 <i>privileged user</i> is the responsible party, the process owner, or a member of a system administration group.
Current user	The <i>current user</i> is the user that completed the most recent task action and is independent of the responsible party.

User	Description
	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 <i>process owner</i> 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	A <i>task owner</i> is a user who is granted privileges for the task's target data.

The Workflow Designer interface

The Workflow Designer interface consists of multiple panels.



- 1 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.
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.


2 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 , List Arrangements , 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 in the 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.

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

3 Additional Workflow Designer functionality is found under **More Commands ...**.

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.
<div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Note:</p> <p>The Export option is available under the Share menu.</p> </div>	
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

Command	Description
	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

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).

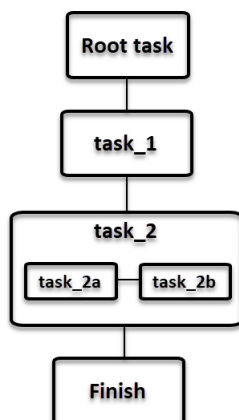
Workflow templates are created from scratch or based on existing templates. Existing templates include the out-of-the-box default templates included with Workflow Designer or those created by your organization.

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:

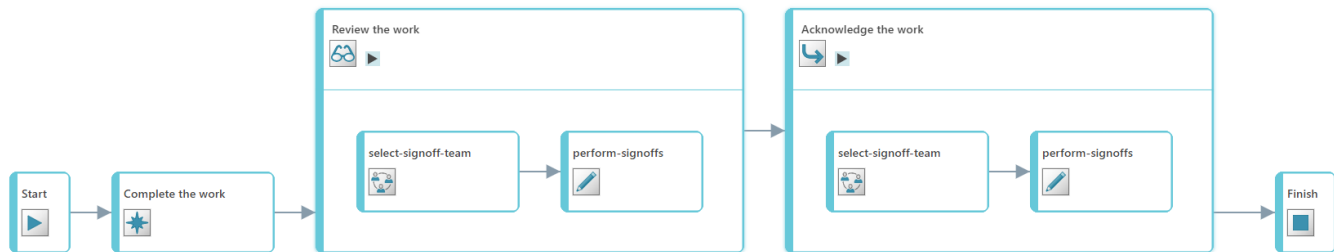
The out-of-the-box templates will always override during upgrades, so if you want to isolate your configuration changes, create Saved As copies of the templates. You can always view the latest changes on the out-of-the-box templates and then decide which changes you want to adopt at your own pace.

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

Core templates

The following table lists the templates and their associated types included with the rich client.

Template name	Task template definition type	Task type value specified in task template	Executing task's real type	Executing task's task type
Process	EPMTaskDefinition	EPMTask	EPMTask	EPMTask
Review Process	EPMTaskDefinition	EPMTask	EPMTask	EPMTask

Template name	Task template definition type	Task type value specified in task template	Executing task's real type	Executing task's task type
Task	EPMTaskDefinition	EPMTask	EPMTask	EPMTask
Review Task	EPMTaskDefinition	EPMReviewTask	EPMTask	EPMReviewTask
Do Task	EPMDoTaskDefinition	EPMDoTask	EPMTask	EPMDoTask
Or Task	EPMTaskDefinition	EPMTask	EPMTask	EPMTask
Add Status Task	EPMTaskDefinition	EPMTask	EPMTask	EPMTask
Change Management Procedure	EPMTaskDefinition	EPMTask	EPMTask	EPMTask
Change Management Item	EPMTaskDefinition	EPMTask	EPMTask	EPMTask

Create a workflow template

1. Go to **More Commands** ... > **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.


Procedure

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 **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 to promote their progression. A path establishes the sequence by which peer-level tasks are run, indicating that the task on the arrow end of the path cannot start until the task on the start end is completed.

Manually created paths, drawn from the predecessor task to the successor task, are explicit paths. Assumed paths are automatically created by the system if no explicit paths have been created from the **Start** node by the time the template is set to the **Available** stage. When you put a workflow template in **Edit** mode and draw a single path from the **Start** node to another task node, assumed path behavior is disabled. The system does not draw assumed paths, even if you leave tasks unlinked and change the workflow template to the **Available** stage. Any unlinked tasks are skipped when a workflow process based on the workflow template is initiated, and no error messages appear.


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

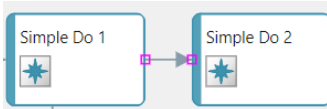
Procedure

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.

- 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**.





- Go to **More Commands**  > **Edit**  > **Online** .

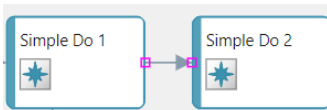
The template is now available to the organization.

Delete paths between tasks

When you delete a task from a template, the system deletes its path along with the task. If you do not reestablish explicit paths among the remaining tasks, the system creates assumed paths.

Procedure

- With the template open, click **Task Palette**  to delete paths in the template.
- To delete the established path type, click the path to display the pink line indicators, and click **Remove** .



The system deletes the selected path.

- Go to **More Commands**  > **Edit**  > **Online** .

The template is now available to the organization.

Failure paths

A failure path gives an alternate course that a workflow can follow in any of the following scenarios:

- A task is rejected.

- The user determines that the task cannot be completed.
- There is an error.

When creating a workflow, each path is configured as either a *success path* or a *failure path*. A failure path must be configured into the workflow process template at design time. A task follows the appropriate path based on the task's outcome. A success path is traversed when a task's state transitions to **Complete** or when a task is promoted and it transitions to a **Skipped** state. A task completes upon the successful execution of the task's handlers on the **Complete** action.

It is important to note that a handler will not execute on a fail path. When a task traverses the failure path, the task state changes to **Failed** (not completed). The complete handlers will only execute when a success path is taken and the task state changes to **Completed**.

Note:

An available option to still execute handlers when a task state changes to failed is to connect the task's failure path to a non-interactive task. The sole purpose of this non-interactive task would be to act as a place where handlers can be added along the failure path, if you want those handlers to execute, such as when a task fails. Subsequently, the non-interactive task's path can then flow into the next task that you want to execute in the workflow.

For example, if a Review task in a workflow gets rejected, including a non-interactive task along the failure path can execute a notify handler (EPM-notify) to communicate that the task was rejected with a failure email.

Backward branching allows a path to be routed backward to some previous task in the workflow process flow, including the **Start** node. Both success and failure paths are capable of branching in a backward direction. Backward branching allows the re-execution of a task with a **Complete** or **Skipped** task state.










To create a failure path, right-click an arrow and select the appropriate failure option. Failure path options display differently for different tasks.

Task	Failure option
Do	Set to Unable to Complete
Review	Set to Reject
Route	Set to Reject
Condition	Set to Unable to Complete
Validate	Set to Error Path
EPM	Set to to Complete

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.
	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.

Symbol	Task	Definition
		Use this task to design workflows around anticipated errors.

Custom tasks

A custom 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. 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 for users to complete.

See [What are custom task templates](#) for more detailed information.

Do tasks

Use the **Do** task to define actions for a user to complete. When this task is performed in a workflow, it displays the required actions to the user in the **Overview** tab of the task.

Note:

Configure the **WRKFLW_task_complete** preference to use single click functionality for a **Do** task. Setting the value to **true**, the **Complete** value is selected by default for the **Done** button.

If you require user authentication before this **Do** task is performed, add the EPM-require-authentication handler to the **Perform** action of the task. When you implement user authentication for this task, a password box appears below the **Comments** box. Users must type their user password in this box before they can click **Apply** and complete the task.

After completing the instructions, the user must select the **Complete** check box. The task does not complete until the user selects the check box. (This task is automatically configured with the EPM-hold handler to stop the task from completing until the check box is selected.) When the user selects the check box, the task sets the handler's argument to **False** and changes the status to **Complete**.

If the task is configured with a failure path the user can select one of the following check boxes:

- **Complete** confirms the completion of the task and continues the workflow down the success path.
- **Unable to Complete** indicates the user is unable to complete the instructions and continues the workflow down the failure path.

Review tasks

Use the **Review** task to route workflow targets (documents, parts, designs, and so on) for review.

Note:

Configure the **WRKFLW_task_complete** preference to use single click functionality for a **Select Signoff Team** review task. Setting the value to **true**, the **Ad-hoc Done** check box is selected by default. This also sets the associated **task_result** property to **Complete** by default.

The task includes two subtasks:

- The **select-signoff-team** subtask requires the workflow initiator to select the users who will perform the review (the signoff team). You can configure this subtask with predefined group/role profiles that the workflow initiator must select or allow the workflow initiator to selector users of his choice in an ad hoc manner.

This subtask uses selection functionality from the Organization application, allowing the selector to search by group/role/user and to select signoff members individually or by project teams or address lists.


- The **perform-signoffs** subtask is then distributed to the selected signoff team, prompting them to review the target objects and signoff.

Caution:

Do not add or delete subtasks from the **Review** task. It may cause an error that prevents the task from executing.

When this task is performed in a workflow, the **perform-signoffs** task displays three options to each signoff team member: **Approve**, **Reject**, and **No Decision**. Selecting either **Approve** or **Reject** performs the task. **No Decision** is the default selection, selecting this option does not perform the task.

If you require user authentication before this **Review** task can be performed, add the EPM-require-authentication handler to the **Perform** action of the task. When you implement user authentication for this task, a password box appears below the **Comments** box. Users must type their user password in this box before they can click **Apply** and complete the task.

If a user manually promotes a **Review** task that has both an **Approve** path and **Reject** path using **Manage**  > **Promote Task** on the primary toolbar in the Inbox, then they must select which path the workflow is to follow at that time.

Add Status tasks

Use the **Add Status** task template to create and add a **Release** status to the target objects of the workflow process.

This template is a visual milestone in the workflow process. There is no action for the user to perform, and therefore, no dialog box associated with the **Add Status** task.

Or tasks

Use an **Or** task template to continue 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. Typically, **Or** tasks are used to unite parallel paths create by:

- True/false condition paths branching from **Condition** tasks.
- Parallel links branching from a single task.

This template is a visual milestone in the workflow process. There is no dialog box associated with the **Or** task.

Acknowledge tasks

Use the **Acknowledge** task to define the **Signoff Team** profiles with which a user complies to assign acknowledgment responsibilities to other users. This template also provides the **perform-signoffs** task for the **Signoff Team** members to complete.

Caution:

- Do not add or delete subtasks from the **Acknowledge** task. It may cause an error that prevents the task from executing.
- Signoff profiles are unavailable for the **Acknowledge** task if it is a subtask within the **Route** task template. The **Route** task does not function properly if signoff profiles are defined for the subtasks. The **Route** task template is designed to be used as an electronic routing sheet, and the workflow initiator assigns specific signoff members.

When this task is performed in a workflow, the **Acknowledge** task displays two decision commands to members of the selected signoff team: **Acknowledged** and **No Decision**. Signoff team members choose one of the above commands to perform the signoff.

If you require user authentication before this **Acknowledge** task is performed, add the EPM-require-authentication handler to the **Perform** action of the task. When you implement user authentication for this task, a password box appears below the **Comments** box. Users must type their user password in this box before they can click **Apply** and complete the task.

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, a user determines the path cannot be completed, or 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 subprocesses (automatic only).

If there are multiple subprocesses, a query runs on the associated subprocesses 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 subprocesses.

If there is only one subprocess 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.

- Configure the task result from the manual **Condition** task's dialog box.

A **Condition** task can be configured to complete either automatically or manually. You need to determine which configuration 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	<p>Add an action handler that sets the task's result to true, false, or a customized value.</p> <p>The simplest way to achieve this is to use the task template's interface to define a condition query at design time; this automatically inserts the action handler. Alternatively, you can create a custom action handler that uses ITK to verify criteria.</p>
Manual Condition task	<p>During design, you do not define a query or add an action handler to the task template.</p> <p>Because no query is defined and no action handler is configured to set the task result, when the workflow</p>

Task	Description
	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.

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.

Route tasks

Use the **Route** task as a router sheet with which a user assigns review, acknowledge and notification responsibilities to specified users.

Note:

Configure the **WRKFLW_task_complete** preference to use single click functionality for a **Select Signoff Team** route task. Setting the value to **true**, the **Ad-hoc Done** check box is selected by default. This also sets the associated **task_result** property to **Complete** by default.


When this task is performed in a workflow, the **Route** task displays three subtasks: **Review**, **Acknowledge**, and **Notify**. The workflow initiator can then assign other users to perform these tasks. The selected users are the signoff team.

Caution:

- Do not add or delete subtasks from the **Route** task. It may cause an error that prevents the task from running.
- Signoff profiles are unavailable for the **Acknowledge** subtask within the **Route** task template. The **Route** task does not function properly if a signoff profile is defined for the **Acknowledge** subtask. The **Route** task template is designed to be used as an electronic routing sheet, and the workflow initiator assigns specific signoff members.

After the **Route** task is performed, the selected signoff team is prompted to perform the **Review** or **Acknowledge** tasks or simply notified of the review through program mail. Notified users do not need to perform any task.

If you want to require user authentication before the **Review** or **Acknowledge** subtasks can be performed, add the EPM-require-authentication handler to the **Perform** action of the subtask (the **perform-signoffs** task of either the **Review** or **Acknowledge** subtasks). When you implement user authentication for either of these subtasks, a password box appears below the **Comments** box. Users must type their user password in this box before they can click **Apply** and complete the task.

If a user manually promotes a **Route** task that has both an **Approve** path and **Reject** path using **Manage**  > **Promote Task** on the primary toolbar in the Inbox, then they must select which path the workflow is to follow at that time.

You can also route or reassign tasks to another user from your inbox, in the same manner as selecting a signoff task. You can select options and designate specific users to notify, acknowledge, or review tasks.

To set up the display of the **Task View** pane, configure the **WORKFLOW_new_route_task_panel** preference. Display choices are **ON** for list box view or **OFF** for option button view.

Validate tasks

The **Validate** task branches a workflow along two or more paths. The path followed is determined by whether specified errors occur during a workflow. Use this task to design workflows around anticipated errors (such as checked out targets), unexpected errors (such as failed scripts or failure of custom handlers), or to track any and all workflow errors.

Configure the **Validate** task by defining one or more *success* and *failure* paths flowing from the task. The success path is followed if no error occurs. The failure path is followed when errors occur.

Note that handlers should be placed on either the **Started** action or the **Completed** action of the Validate task.

When errors occur, you determine if the failure path is followed when:

- Any error occurs.
- Only when an error you specify on a list of error codes occurs.

Note:


In the context of the **Validate** task, *workflow error* means any error generated by a workflow handler.

Configure the task to follow a failure path by **pairing** a workflow handler and an error code. Place a handler to be validated on the **Validate** task and then add the respective error code to the path's error list (or set the path to fail on *any* error).

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 *Start an ad hoc 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.

This pseudo folder **Independent Sub-Workflows** displays subworkflows identified from the property **epm_subprocesses**. Subworkflows are presented in no particular order, nor can their order be controlled.

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.

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	

Add the EPM-create-sub-process handler

Add the **EPM-create-sub-process** handler to a task in a workflow template to start a subworkflow.

Procedure

- Select a template from the list.
- Click **Start Edit** .
- Select one of the following:
 - Keep the template online during the editing process.
 - Keep the template offline during the editing process.

4. Select the task to contain the subworkflow.
5. Click the **Handlers** tab.
6. In the handlers table on the left, select the trigger group to add the handler to: **Started** or **Completed**.
7. Under **New Handler** on the right, click **Action** as the handler type.
8. Select the **EPM-create-sub-process** handler from the drop-down list.
9. Select the name of the template to contain the subworkflow from the drop-down list.
10. Click **Add**.
11. Click **End Edit**  to save your new subworkflow to the template.

Creating subworkflows for assemblies

In workflows that contain assemblies, there are various arguments you can use with the EPM-create-sub-process action handler to create subworkflows for components of the assemblies.

Argument	Behavior
-process_assembly	Searches for assemblies in the target, reference, or all (as specified by the -from_attach argument) and creates subworkflows for each component.
-depth	Specifies the depth to which the assembly is traversed.
-rev_rule	Specifies the revision rule applied to the assembly.
-include_related_type	Creates subworkflows only for assembly components of the types specified in this argument.
-exclude_related_type	Does not creates subworkflows for assembly components of the types specified in this argument.

Note:

The **-include_related_type** and **-exclude_related_type** arguments can be used in conjunction with each other. If used in conjunction, the **-include_related_type** argument takes precedence;

first the objects are processed against **-include_related_type** and then processed against **-exclude_related_type**.

Creating subworkflows for related objects

There are various arguments you can use with the EPM-create-sub-process action handler to create subworkflows for related objects of target and reference data.

Argument	Behavior
-relation	Creates subworkflows for each object attached by the specified relation to the target or reference object. (Specify a particular target, or reference object, or all, using the -from_attach argument.)
-include_related_type	Creates subworkflows only for related objects of the type(s) specified in this argument.
-exclude_related_type	Does not creates subworkflows for related objects of the type(s) specified in this argument.

Note:

The **-include_related_type** and **-exclude_related_type** arguments can be used in conjunction with each other. If used in conjunction, the **-include_related_type** argument takes precedence; first the objects are processed against **-include_related_type**, and then **-exclude_related_type**.

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, go to **More Commands** ... > **New** ✨ > **Create Workflow Template** 🏠.
2. Click **Task** as the **Template Type**.
3. Enter a unique **Name** and **Description** (optional).
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. 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, go to **More Commands > New**  **> Create Workflow Template** .
2. Click **Task** as the **Template Type**.
3. Enter a unique **Name** and **Description** (optional).
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 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

The file **performTaskPanelConfiguration.json** 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 === 'EPMPPerformSignoffTask' ||
        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 !==
        'EPMPPerformSignoffTask') && modelObject.modelType.typeHierarchyArray.indexOf('EPMTask'))"
    }
  }
}
```

```

    >= 0) "
    }
  },
  "performTaskToolAreaContribution":
  {
    "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 === 'EPMPPerformSignoffTask' ||
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 !==
'EPMPPerformSignoffTask') && 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="il8n.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 requires:

- Creation of a custom 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'"
    }
  },
  "performTaskToolAreaContribution":
  {
    "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 the default view displays 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' "
    }
  },
  "performTaskToolAreaContribution":
  {
    "Awp0EPMDoTaskPerform1" : {
      "id": "Awp0EPMTaskPerform",
      "priority": 2,
      "performPanelId": "Awp0EPMConditionTaskPerform",
      "condition": "modelObject.type === 'EPMDoTask' "
    }
  }
}
```


The provided task type 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 registered type is **Awp0EPMTaskPerform**.

Use the custom task created in Active Workspace in the rich client

If you create a custom task in Active Workspace and plan on using that task in the 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

Determining which editing options to use

Perform edits on existing workflow process templates by selecting the template to be edited and clicking the **Edit Mode**  button.

Consider the following questions before editing a workflow template.

Editing task	Description
Edit offline or online?	<p>Offline editing prevents users from accessing the workflow template while you edit. Use this option when you do not want the old version of the workflow template available for use until your edits are complete.</p> <p>Online editing allows users to initiate workflows based on the old version of the workflow template, while you edit a copy of the same template. When you switch the edited version to the Available stage, the older copy is overwritten; only the edited copy remains available from the interface.</p>
Apply edits to running workflow processes?	<p>After editing a workflow template, you can apply the edits to all active processes that are based on the template. When you select the Set Stage to Available check box, the Apply Template Changes dialog box asks whether to apply the edits to all active workflow processes based on the template.</p> <p>Select the Apply template changes to all active workflow processes check box to update each active workflow process based on the workflow template as follows:</p> <ul style="list-style-type: none">• If the edits in the workflow template occur <i>later</i> in the workflow than the active workflow process has reached, the edits are applied to the workflow.• If the edits in the workflow template occur <i>earlier</i>, and the active workflow has already passed the place where the edits were made, the edits do not take effect unless the task or path is re-run (using backward branching or loops) or the task is demoted.• If the edits in the workflow template impact an active task, the edits are applied after the task completes and take effect only if the task is re-run.

Editing task	Description
Which workflow components can be edited?	<ul style="list-style-type: none"> • If the edits delete the currently active task, the next task is started. You can edit any aspect of the workflow process template, including: <ul style="list-style-type: none"> • Changing the template name • Adding and removing tasks • Adding, deleting, redrawing, and resetting flow paths • Adding, deleting, and resetting handlers, attributes, task attributes, and attachments

Offline versus online editing

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.

Identify templates as non-modifiable

Users identified by the **Awp0IsPrivilegedWorkflowTemplateAuthor** BMIDE condition as privileged can specify templates as requiring administrative privileges for modification so that other users cannot modify them in the Workflow Designer on Rich Client.

Using the **Edit** mode in Workflow Designer, you can modify existing templates or templates you are actively developing to be read-only.

Procedure

1. **Create a new template**, or select a workflow template from the list and click **Start Edit** .
2. Ensure that no task is selected in the template. On the **Properties** tab, click the **Require admin privilege for modification** check box.


Show Task in Process Stage List:	<input checked="" type="checkbox"/>
Process in Background:	<input type="checkbox"/>
Select Participants from Workflow:	<input type="checkbox"/>
Require Task Confirmation on Complete:	<input type="checkbox"/>
Require admin privilege for modification:	<input type="checkbox"/>


Once saved, templates are non-modifiable and can only be edited by privileged users who are identified based on the BMIDE condition **Awp0IsPrivilegedWorkflowTemplateAuthor**. All other users cannot access the **Start Edit** button once the read-only template is open in the Workflow Designer on Rich Client.

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 **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.

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


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

- 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 **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 **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 rich client or Active Workspace. This also lets you view legacy template layouts as they were originally designed in the 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, 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 **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. 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.

```
((o !=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_alterate_procedures** preference is set to **any** or **Assigned**, there is no restriction on what type of target can be added.
- If the **CR_allow_alterate_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_alterate_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_alterate_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 **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.

Properties Access Handlers Assignments Notifications

CR Process


Instructions:

Workflow for processing a Change Request Revision.

Change Request Revision

- ☒ Change Request Revision (ChangeRequestRevision)
- ☐ Change Request Revision Master (ChangeRequestRevision Master)
- ☐ Generic Change Request Revision (GnChangeRequestRevision)
- ☐ Generic Change Request Revision Master (GnChangeRequestRevision Master)

Change Request Revision (ChangeRequestRevision)

- 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. Go to **More Commands ...** > **Import/Export**  > **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. Go to **More Commands ... > Import/Export**  **> 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 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.

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



3. Select **Settings**  > **Arrange** to customize the workflow criteria available for comparison.
4. To remove the selections, deselect each template.

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. Go to **More Commands**  > **Edit** > **Delete** .
3. Choose **Delete** from the dialog box to permanently remove the template.

Distributing workflow templates using Multi-Site Collaboration

Replicate a workflow template

You can distribute your workflow templates to different Teamcenter sites by replicating templates using **Multi-Site Collaboration**. You can replicate your workflow templates, including those under construction, on several Teamcenter sites by using the **data_share** utility and update them with the **data_sync** utility. You cannot edit the replicas, only the template at the owning site. Also, handlers attached to the templates must exist at all sites where the templates are replicated.

Procedure

1. If necessary, **create the template** that you want to replicate.
2. Run the `data_share` utility with the following arguments:


```
data_share -u=user-id -p=password -g=group -f=send -site=remote-site-name1  
-name=workspace-object-class=class-name
```

For example, if you want to replicate the **demotemplate** workflow template at the **teamcentersite2** site, run the following utility command (the required logon information is omitted from the example):

```
data_share -f=send -site=teamcentersite2 -name=demotemplate  
-class=EPMTaskTemplate
```

Note:

- If you want to transfer ownership to the specified site, add the **-transfer** argument to the command.
- If you want to import the template at another site to the current site, change the **-f** argument to **-f=remote_import**.
- If you want to replicate the template at more than one site, add more **-site** arguments to the command.
- If you want to replicate several templates, type the template names in a text file and replace the **-name** and **-class** arguments with the **-filename** and **-classoffile** arguments, respectively.

The replicate template appears at the new site with the  symbol.

Synchronize replicated templates

You can use Multi-Site Collaboration to synchronize replicated templates at remote sites.

Procedure

1. Update the template at the owning site that is replicated at another site.

Note:

If you want *active* workflows to be based on the synchronized template to be updated at the replica site, set the **WRKFLW_multisite_apply_template_changes** preference to **true**.

2. Run the `data_sync` utility with the following arguments:

```
data_sync -u=user-id -p=password -g=group -f=sync -site=remote-site-name1 -class=class-name -update
```

For example, if you changed the **demotemplate** workflow template and wanted to update the replica at the **teamcentersite2** site, run the following utility command (the required logon information is omitted from the example):

```
data_sync -f=sync -site=teamcentersite2 -class=EPMTaskTemplate -update
```



Note:

If you want to synchronize the template at more than one site, add more **-site** arguments to the command.

The replicate template is updated at the specified sites.

6. Managing Task Behavior

Delete a task

1. Open the template containing the task you wish to delete, and click **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 **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.

Access Control Lists

Access control lists (ACLs) contain a list of privileges granted, denied, or not set for each accessor. In the Workflow Designer, you can create and apply **Workflow** ACLs or select and apply **System** ACLs to a task.

- A **Workflow** ACL is specific to a workflow. Objects can use a Workflow ACL to define access and determine what operations can be performed on it. For example, you could create a workflow ACL when you want to provide permission for a user to promote a task, or for approvers to gain write access to the target object. Workflow ACLs can stay in effect for the task until the same workflow sets another ACL later in the process or the workflow is completed.
- A **System** ACL is one that has been already defined outside of Workflow Designer. It can be applied to the task and all subsequent tasks in the workflow unless it is changed by another instance of the **EPM-set-rule-based-protection** handler or the process is completed. You can select and apply a **System** ACL in the Workflow Designer, but information about existing System ACLs is read-only. To create a System ACL, open **Access Manager** by clicking its tile on the **Home** page.

How ACLs work with Access Manager

In Workflow Designer, the **Access** tab lets you create and apply a Workflow ACL or configure a task with a specific System ACL for Access Manager-enabled systems.

ACLs work in parallel with the **EPM-set-rule-based-protection** handler, which 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 the handlers execute.

For example, a desired configuration may be to have all of the following applied 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.


In Workflow Designer, the **Access** tab lets you create and apply a Workflow ACL or configure a task with a specific System ACL for Access Manager-enabled systems.

Create a Workflow ACL

A Workflow ACL is specific to a workflow. It can stay in effect for the task until the same workflow sets another ACL later in the process or the workflow is completed. In Workflow Designer, use the **Access** tab to create a Workflow ACL.

Procedure

1. Open the template that contains the tasks you want to modify, and click **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 **Access** tab.
4. Under **Access Type**, click **Workflow**.

5. Under **Named ACL**, click **Manage ACL** .

The **Manage ACL** panel opens.

6. Under **Create New ACL**, enter the new **ACL Name** and click **Create**.

The new workflow ACL instance is created but accessors and privileges are not assigned yet.

7. Ensure the correct task is selected in the workflow for the ACL. If no task is selected, the ACL is applied to the root task.

8. Select the new ACL from the **ACL Name** drop-down list.



9. Under **Object Access Control List**, click **Add** .





A new row of access control entries (ACEs) are available to define the privileges for the new workflow ACL.

10. Select the Accessor Type from the drop-down list.

11. (Optional) Select the **Accessor** from the drop-down list. *Accessors* are collections of users who share certain common traits, such as membership in the group that owns the object or membership in the project team. Some Accessors Types do not require an Accessor.

For example, if you chose the **Accessor Type** of **Participant**, the list of **Accessors** would display every type of Dynamic Participant available.

12. Set privileges by clicking the cell corresponding to the privilege you want to set, and choose to grant privileges  or deny privileges . Hover over the cell to identify the icons.

Click **Allow All**  to grant access for all entries or **Deny All**  to restrict access to all entries. **Reset**  returns the selected row to the last saved values for the privileges, while **Clear All**  clears all of the cells for the selected row.

At least one privilege must be granted or denied for the **Object Access Control List** table.

Note:

Whenever possible, do not explicitly set privileges. Leaving privileges unset allows rules to accomplish focused objectives by allowing objects and accessors to filter through rules that do not apply to them.

13. Click **Save**.

The ACL is applied to the selected task.

Workflow

Show Task Information | Diagram Settings | Apply Layout | Task Palette | Remove | Save Layout

Online | End Edit | Full Screen

TASK TYPES

- Acknowledge Task
- Add Status Task
- Condition Task
- Do Task
- Or Task
- Review Task
- Route Task
- ☒ Task
- Validate Task

```

graph LR
    Start([Start]) --> Complete[Complete the work]
    Complete --> Review[Review the work]
    Review --> Acknowledge[Acknowledge the work]
    Acknowledge --> Finish([Finish])
  
```

Properties | **Access** | Handlers | Notifications | Assignments | List Arrangements

OBJECT ACCESS CONTROL LIST

Accessor Type	Accessor	View	Edit	Delete	Share	Move	Copy	Paste	Print	Download	Upload	Link	Unlink	Reset	Clear
Responsible Party		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Save

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

Properties | **Access** | **Handlers** | Notifications | Assignments | List Arrangements

Name: Complete the work

Trigger	Arguments
Assigned	
Started	
EPM-set-rule-based-protection -acl=New ACL	
Performed	
Completed	
Promoted	
Suspended	
Resumed	
Aborted	
Undo	

Note:




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





Modify a Workflow ACL

A **Workflow** ACL is specific to a workflow. Once you create a **Workflow** ACL, you can modify its settings.

See [Create a Workflow ACL](#) for more information about how to create an ACL.

Procedure

1. Open the template that contains the tasks you want to modify, and click **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. Modify privileges by clicking the cell corresponding to the privilege you want to edit, and choose to grant privileges  or deny privileges . Hover over the cell to identify the icons where necessary.

Click **Allow All**  to grant access for all entries or **Deny All**  to restrict access to all entries. **Reset**  returns the selected row to the last saved values for the privileges, while **Clear All**  clears all of the cells for the selected row.

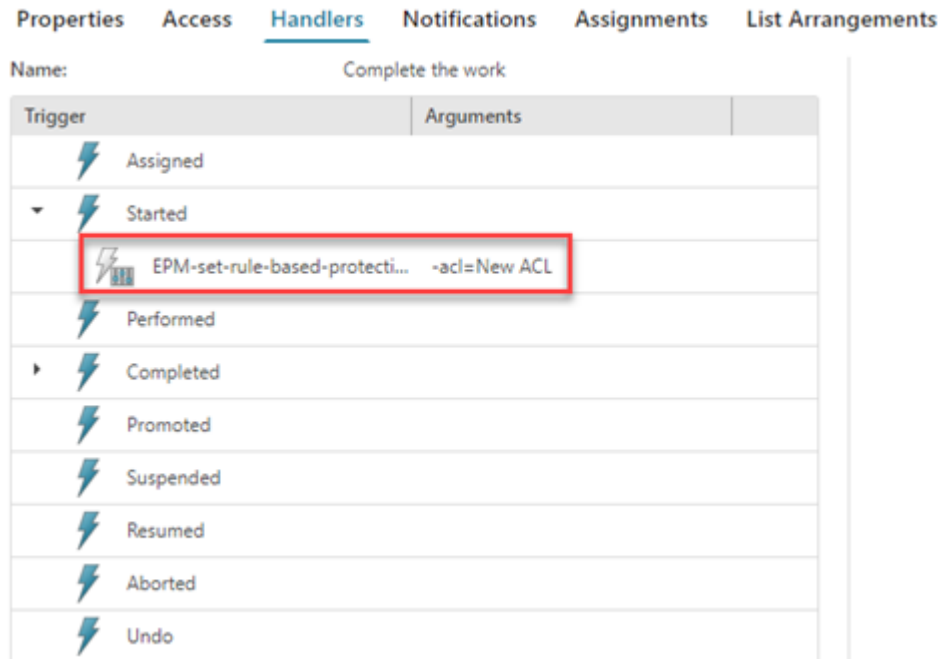
At least one privilege must be granted or denied for the **Object Access Control List** table.

Note:

Whenever possible, do not explicitly set privileges. Leaving privileges unset allows rules to accomplish focused objectives by allowing objects and accessors to filter through rules that do not apply to them.

6. Click **Save** and the **Workflow** ACL is updated.



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




Delete a Workflow ACL

When you delete a Workflow ACL from a task in a template, the Workflow Designer automatically deletes the associated **EPM-set-rule-based-protection** handler from the task once the edits are saved.

Procedure

1. Open the template that contains the tasks you want to modify, and click **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 contains the Workflow ACL to delete.
4. Click the **Access** tab.
5. Under **Named ACL**, click **Manage ACL** .

The **Manage ACL** panel opens.

6. Under **ACL**, select the ACL to delete from the list and click **Delete** .
7. At the prompt, click **Delete**.

A message confirms that the ACL was deleted. The associated **EPM-set-rule-based-protection** handler for the task is also removed from the **Handlers** tab. Any other tasks that use the deleted ACL will have the name removed from the **Access** tab, but the handler remains.

Import a template that includes an ACL on a task

When you import a template that includes an ACL on a task(s), the ACL name, or instance, imports but its associated accessors and privileges do not. The associated **EPM-set-rule-based-protection** handler remains on each task in the imported template, but you must create and redefine the ACL details for the task on the **Access** tab for the ACL to apply to all tasks. Once you create and redefine the privileges for one imported ACL, the other tasks in the template that included the ACL are automatically populated with the same details.

Procedure

1. **Import the template** that contains the ACL.

2. Select a task that contains an ACL and click the **Access** tab.

Note that the ACL is not selected on the task once it imports.

3. Under **Named ACL**, click **Manage ACL** .

The **Manage ACL** panel opens.

4. Under **Create New ACL**, enter the **ACL Name** and click **Create**. Use the same ACL name as that in the imported handler.

The new workflow ACL instance is created but accessors and privileges are not assigned yet.



5. Select the newly-created ACL from the **ACL Name** drop-down list.

6. Under **Object Access Control List**, click **Add** .

A new row of access control entries (ACEs) are available to define the privileges for the new workflow ACL.

7. Select the Accessor Type from the drop-down list.

8. Select the **Accessor** from the drop-down list.

9. Set privileges by clicking the cell corresponding to the privilege you want to set, and choose to grant privileges  or deny privileges . Hover over the cell to identify the icons.


10. Click **Save**.

The ACL is applied to the selected task and all subsequent tasks that the ACL applied to in the imported template.

Specify an existing ACL for a task

In Workflow Designer, use the **Access** tab to configure a task with a specific ACL for Access Manager-enabled systems. You can specify to use either a **Workflow** or **System** ACL.

Procedure

1. Open the template that contains the tasks you want to modify, and click **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 apply the ACL to.
4. Click the **Access** tab.
5. Click the **type of ACL** to apply:
 - **System**
 - **Workflow**
6. Under **Named ACL**, click the **ACL Name** drop-down list to select a specific ACL.
7. Click **Save** 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.

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.

Add task handlers

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

In the left pane, the table lists any handlers assigned to the selected task. Handlers are listed under their specific trigger group to represent the execution order of action and rule handlers. To see any handlers currently assigned under a specific trigger, click expand ▼.

Name: Review the work

Trigger	Arguments
Promoted	
Suspended	
Resumed	
Aborted	
Undo	
EPM-demote	

- Select a trigger group to add the handler to.
- On the **New Handler** panel on the right, click the handler type you want to create: **Action** or **Rule**.
- All of the applicable handlers for the trigger group are available in the drop-down list. The handler list is not case-sensitive. This list may contain custom handlers as well as handlers that are included with other solutions.

Select a handler.

The mandatory arguments display based on the handler selection.

Properties Access **Handlers** Notifications Assignments List Arrangements

Name: Template 011923PK

Trigger	Arguments
Assigned	
Started	
Performed	
Completed	
Promoted	
Suspended	
Resumed	
Aborted	
Undo	

New Handler

Handler Type: ☒ Action ☐ Rule

Name: [Help](#)

[Add Arguments](#)

Argument: Value(s): [Delete](#)

Add

Click **Help** (?) to launch the handler documentation.

- Add a new argument row by clicking **Add Arguments** (+).
- Select the new argument from the drop-down list.
- Enter any 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**.

Expand the trigger group to view the new handler in the table. Once added, you can highlight the new handler and click **Cut** ✂, **Copy** 📄, or **Delete** ⊖. To make any additional changes to the argument, highlight the row and edit the handler in the right-side **Information** panel.

Remove task handlers

1. Open the template that contains the tasks you want to modify, then click **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 on the left, highlight the action or rule handler that you want to delete under the trigger group and click **Delete** ⊖.

Update task handlers

1. Open the template that contains the tasks you want to modify, then click **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 on the left, highlight the action or rule handler that you want to edit.

To make any additional changes to the argument(s), edit the handler in the **Information** panel on the right. You can add additional arguments, or enter a different value for the argument in the **Value(s)** field, and click **Save**.

Other available editing options include:

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. Templates do not need to be in edit mode to copy a handler.

A handler 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.

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.

View 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.
3. Click the **Handlers** tab to view the **Rule** and **Action** handlers assigned to the task.

In the left pane, the table lists any handlers assigned to the selected task. Handlers are listed under their specific trigger group to represent the execution order of action and rule handlers.

Click the expand icon to see any handlers currently assigned under a specific trigger group ▼.

To view handler information for the root task, click the initial **Start** task.



The **Information** panel on the right displays the handler details.

The screenshot shows the Workflow Designer interface with the **Handlers** tab selected. The left pane displays a list of triggers and their arguments:

Trigger	Arguments
Assigned	
Started	
EPM-attach-related-objects	-relation=IMAN_specification
EPM-attach-related-objects	-relation=PSBOMViewRevision
EPM-assign-team-selector	-assignee=\$PROCESS_OWNER
EPM-auto-assign-rest	
Performed	
Completed	
Promoted	

The right pane shows the **Information** panel for the selected handler **EPM-attach-related-objects**:

- Name:** EPM-attach-related-objects
- Argument:** Value(s):
- attachment:** target
- relation:** IMAN_specification
- Hints:** Please make sure to add either of these "-relation, -property, -lov".

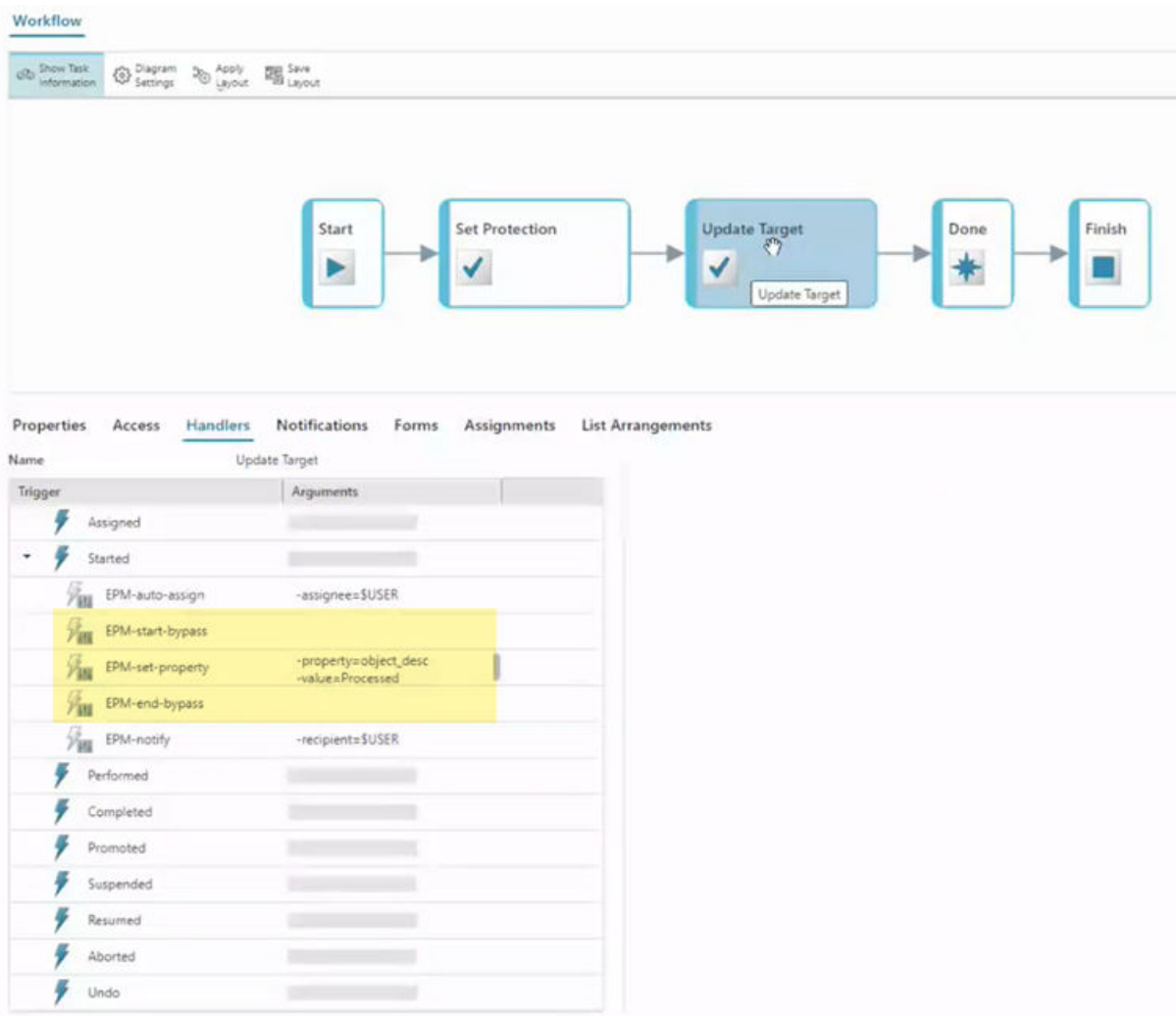
A **Save** button is located at the bottom right of the Information panel.

Bypass handlers

Bypass handlers (**EPM-start-bypass** and **EPM-end-bypass**) prevent user access errors by letting a select set of handlers run with system access. Administrators can define system authority for any handlers wrapped in between **EPM-start-bypass** and **EPM-end-bypass**, providing an alternative to running all handlers with either all or no system access.

EPM-start-bypass starts the bypass process, while **EPM-end-bypass** ends bypass. Any handlers in between grant users the access they need to make necessary updates.

Any number of handlers can be included between the bypass handlers. However, these handlers are only operative within a given action, although they can be configured throughout multiple actions. Here, the **EPM-set-property** handler is provided system access in the **Started** action, as it is included in between the two bypass handlers.



Consider the following when using the bypass handlers:

- If bypass is enabled, then handlers run with system authority and **EPM-start-bypass/EPM-end-bypass** have no effect.
- The **WRKFLW_access_level_for_handlers_execution** preference has precedence over bypass subset handlers when it is set to **system** access. When system authority is enabled for all handlers, the **EPM-start-bypass/EPM-end-bypass** pair has no effect.
- A subworkflow inherits bypass from its parent workflow when the **EPM-create-sub-process** handler is executed within the bypass handlers' scope.
- Both handlers must be included and work as a pair when executing the bypass process.
- If the **EPM-end-bypass** handler is configured but the **EPM-start-bypass** is not configured, then bypass is not enabled and the **EPM-end-bypass** handler has no effect.




- If the **EPM-start-bypass** handler is configured more than once before the **EPM-end-bypass** handler is configured, then during execution, the first occurrence of the **EPM-start-bypass** handler starts bypass and the other **EPM-start-bypass** handlers have no effect. Likewise, if the **EPM-end-bypass** handler is configured more than once following an **EPM-start-bypass** handler, then the first occurrence of the **EPM-end-bypass** handler ends bypass and the other **EPM-end-bypass** handlers have no effect.

See the Handlers guide for more information about the **EPM-start-bypass** and **EPM-end-bypass** handlers.

Assign users to tasks

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


The type of task determines the number of users that can be assigned. Most tasks require a single user, but multiple users can be assigned to a Review, Acknowledge, or a Route task (see the following section).

1. Open the template that contains the tasks you want to modify, and click **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. Select either **Key Roles**, **Teams**, or an individual **User/Group Member** from the drop-down list, then click **Add** .

- **Key Roles**

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

To add a key role, search for and select the key role to add for the task, then click **Add**.

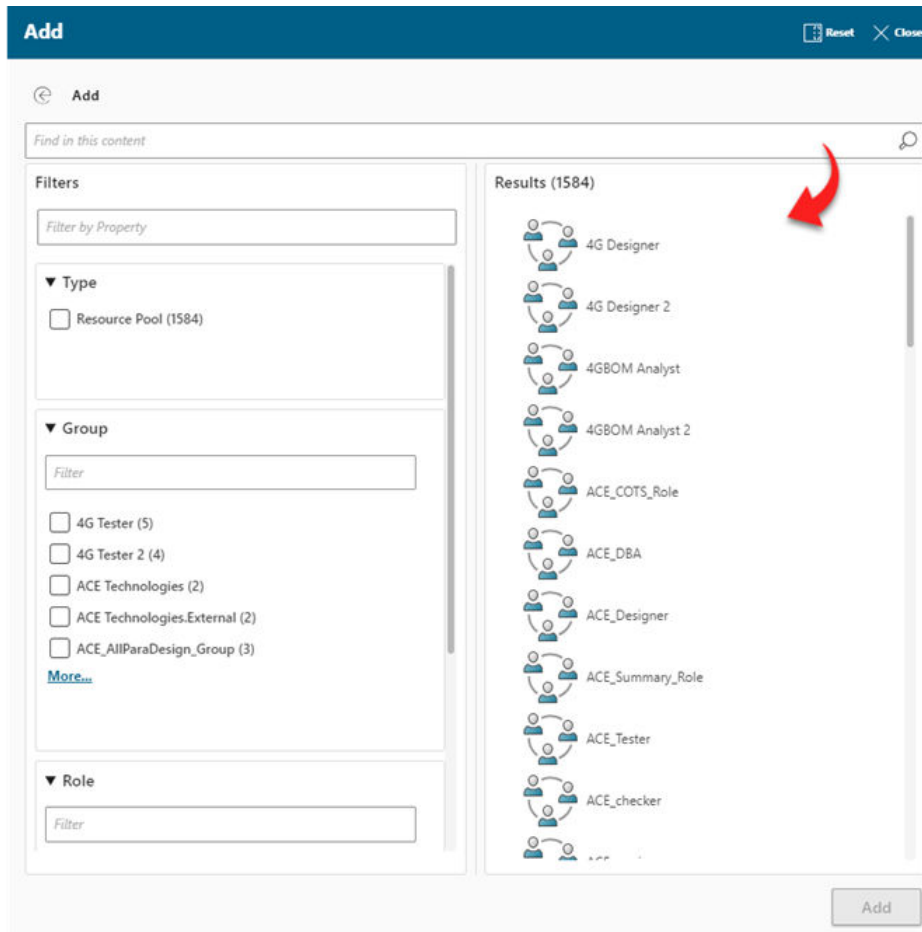
To remove your selection, click the key role and click **Remove** .

- To add **Teams** (predefined teams to choose from) or a **User/Group Member** (available participants in a specific group or role):

- a. Type the name of the team or user to assign and press **Enter** or click **Search** 🔍 to see results. You can enter a partial name or wildcard characters.
- b. For additional search filters, click the **Filter** icon 🗑️.

You can also display the filters and increase the size of the **Add** panel by dragging the left side of the panel to the left.

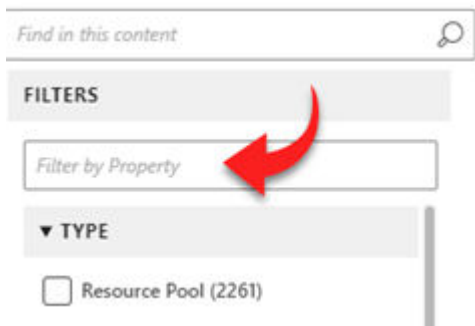
Results now display on the right.



Use the keywords **userid**, **group**, or **role** to narrow your search. Combine keywords by including a space in between (such as **role:checker userid:ed**), and for any term that contains a space, use quotations (**group:"Workflow Reports"**).

Once a search is performed and the results display, this top-most search box can be used again to additionally narrow those results.

- c. Filters are defaulted to **Type**, **Group**, or **Role**. If the user is assigned to a project, the **Project** filter is available. A search box narrows your search to a specific property.



You can also search within each property.

- d. Make your selection and click **Add**.

To remove your selection, highlight the user/team and click **Remove** ⊖.

7. Click **Add**.

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.

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.



Procedure

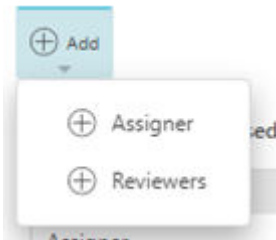
1. Open the template that contains the Review task you want to modify, then click **Start Edit** .
2. Select one of the following:
 - Keep the template online during the editing process.
 - Keep the template offline during the editing process.
3. Click the Review task that you want to modify.
4. Click the **Assignments** tab.
5. Under **Participation**, specify the type of approval quorum: **Percent** or **Numeric**.
6. 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.
7. Click **Require full participation** if you want all users to review and comment before the task progresses.

Add assigners and reviewers to a Review task

Multiple users can be assigned to a Review, Acknowledge, or a Route task.

Procedure

1. Open the template that contains the Review task you want to modify, then click **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 modify.
4. Click the **Assignments** tab.
5. Under **Assignments**, click **Add**  and select either **Assigners** or **Reviewers**.



For both the **Assigner** and **Reviewer** role, you can add users from the list by **Key Roles**, **Teams**, or (as individual) **User/Group Members**.

Key Roles


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

Teams


Predefined teams to chose from.

User/Group Member




Available participants in a specific group or role.

6. For the **Assigners**:
 - a. Select either **Key Roles**, **Teams**, or an individual **User/Group Member** from the drop-down list.
 - b. Click **Add** .

- c. To add a **Key Role**, search for and select the key role to add for the task, then click **Add**.

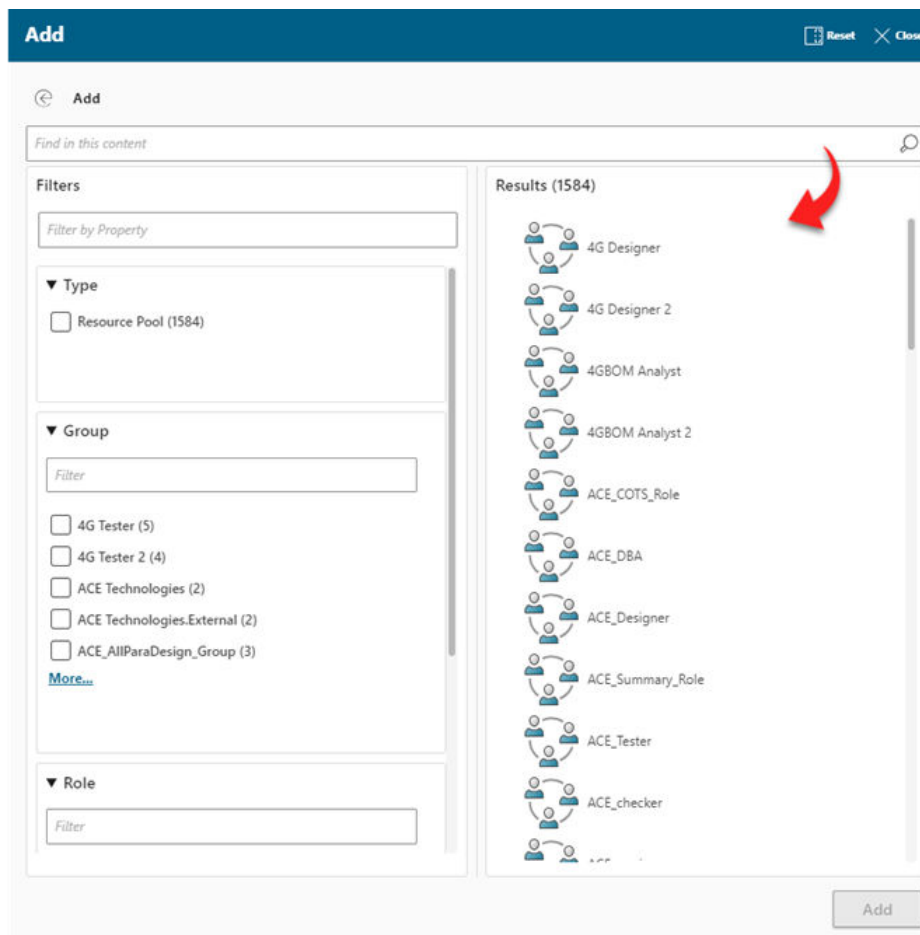
To remove your selection, click the key role and click **Remove** .


- d. If you select **Teams** or **User/Group Member**:

- A. Click **Add** .
- B. Type the name of the team or user to assign and press **Enter** or click **Search**  to see results. You can enter a partial name or wildcard characters.
- C. For additional search filters, click the **Filter** icon .

You can also display the filters and increase the size of the **Add** panel by dragging the left side of the panel to the left.

Results now display on the right.

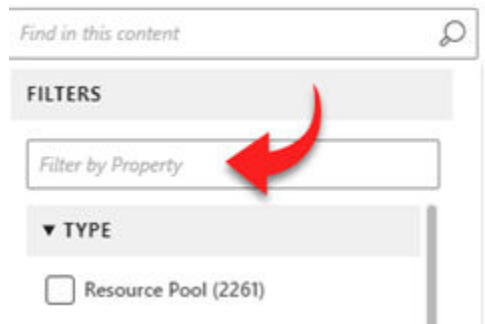


You can click **Reset**  to reset the panel to its default location and size.

Use the keywords **userid**, **group**, or **role** to narrow your search. Combine keywords by including a space in between (such as **role:checker userid:ed**), and for any term that contains a space, use quotations (**group:"Workflow Reports"**).

Once a search is performed and the results display, this top-most search box can be used again to additionally narrow those results.

- D. Filters are defaulted to **Type**, **Group**, or **Role**. If the user is assigned to a project, the **Project** filter is available. A search box narrows your search to a specific property.



You can also search within each property.

- E. Make your selection and click **Add**.

To remove your selection, highlight the user/team and click **Remove** ⊖

7. For the **Reviewers**:

- a. Select either **Key Roles**, **Teams**, or an individual **User/Group Member** from the drop-down list.
- b. Click **Add** ⊕.
- c. Filters act the same as **Assignee**; make your selection and click **Add** ⊕.
 - For **Key Roles** and **User/Group Member**, the option **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.
 - For **Teams**:
 - A. Enter the **Number of Reviewers** to review the task.
 - B. 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.

- d. 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. Once you select your assignees/reviewers, click **Add**.



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 **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.

a. Select either **Key Roles**, **Teams**, or an individual **User/Group Member** from the drop-down list.

- **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.

b. Click **Add** ⊕.

c. To add a **Key Role**, search for and select the key role to add for the task, then click **Add**.

To remove your selection, click the key role and click **Remove** ⊖

d. If you select **Teams** or **User/Group Member**:

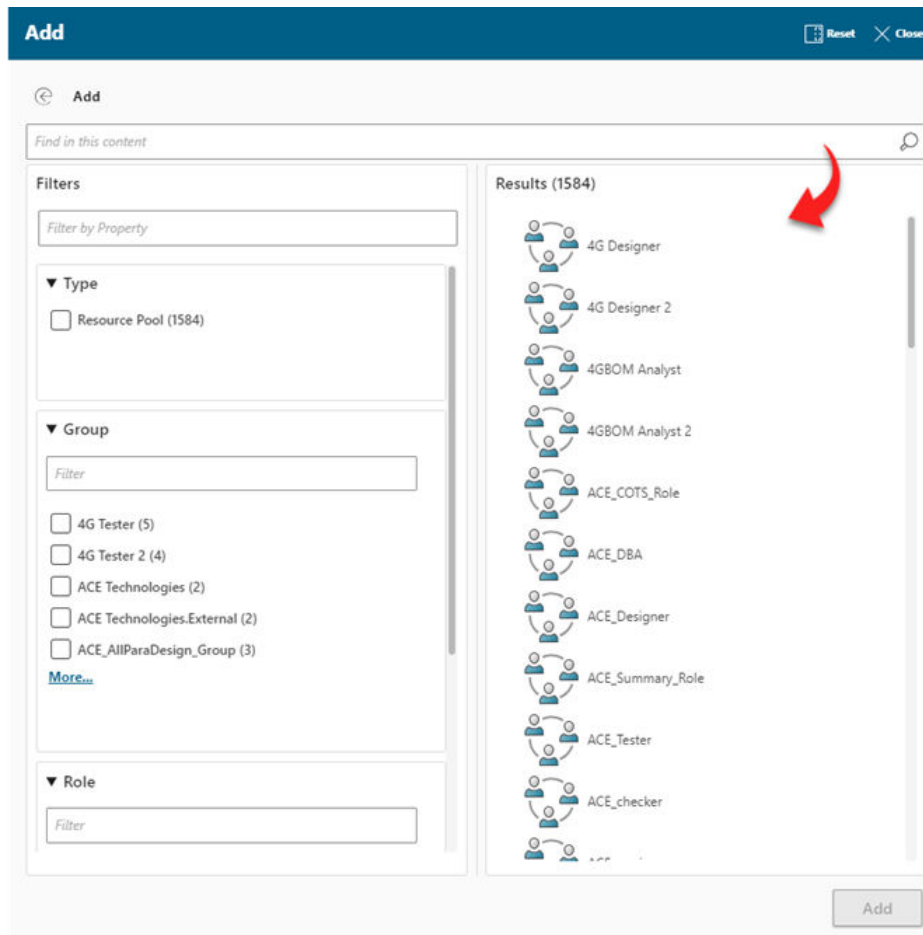
A. Click **Add** ⊕.


B. Type the name of the team or user to assign and press **Enter** or click **Search** 🔍 to see results. You can enter a partial name or wildcard characters.

C. For additional search filters, click the **Filter** icon 🗑.

You can also display the filters and increase the size of the **Add** panel by dragging the left side of the panel to the left.

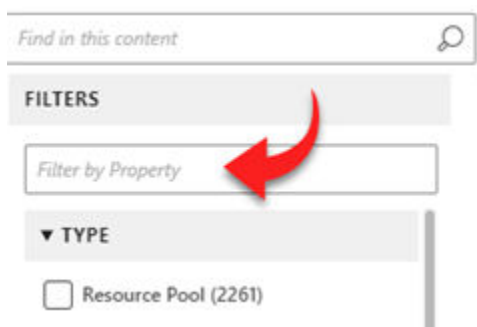
Results now display on the right.



You can click **Reset**  to return the panel to its default location and size.

Once a search is performed and the results display, this top-most search box can be used again to additionally narrow those results.

- D. Filters are defaulted to **Type**, **Group**, or **Role**. If the user is assigned to a project, the **Project** filter is available. A search box narrows your search to a specific property.



You can also search within each property.

- E. Make your selection and click **Add**.

To remove your selection, highlight the user/team and click **Remove** ⊖

11. Once you've selected the recipient(s) to be notified, click **Add**.

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 **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 **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	<ul style="list-style-type: none"> • 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.

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

14. Click **Require admin privilege for modification** to ensure templates are non-modifiable and can only be edited by a **dba** user.


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 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**.

Prerequisites

Use the Business Modeler IDE to add a **decision label** for the **Review** or **Acknowledge** task. For more information, see [Add custom decision labels](#).

Procedure

1. Open the template that contains the tasks you want to modify, and click **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 selection adds decision buttons to the workflow task.

Add a release status to an Add Status task

Add Status tasks create and add a release status to the target objects of the workflow. A release status is a visual milestone in a workflow.

Status tasks have optional properties for task behavior.

Procedure

1. Open the template that contains the tasks you want to modify, and click **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	<p>Add an action handler that sets the task's result to true, false, or a customized value.</p> <p>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.</p>
Manual Condition task	<p>During design, you do not define a query or add an action handler to the task template.</p> <p>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.</p> <p>Custom conditions can also appear as manual condition options and appear as buttons in the Condition dialog box.</p>

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 a BMIDE condition to a Condition task

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

Prerequisites

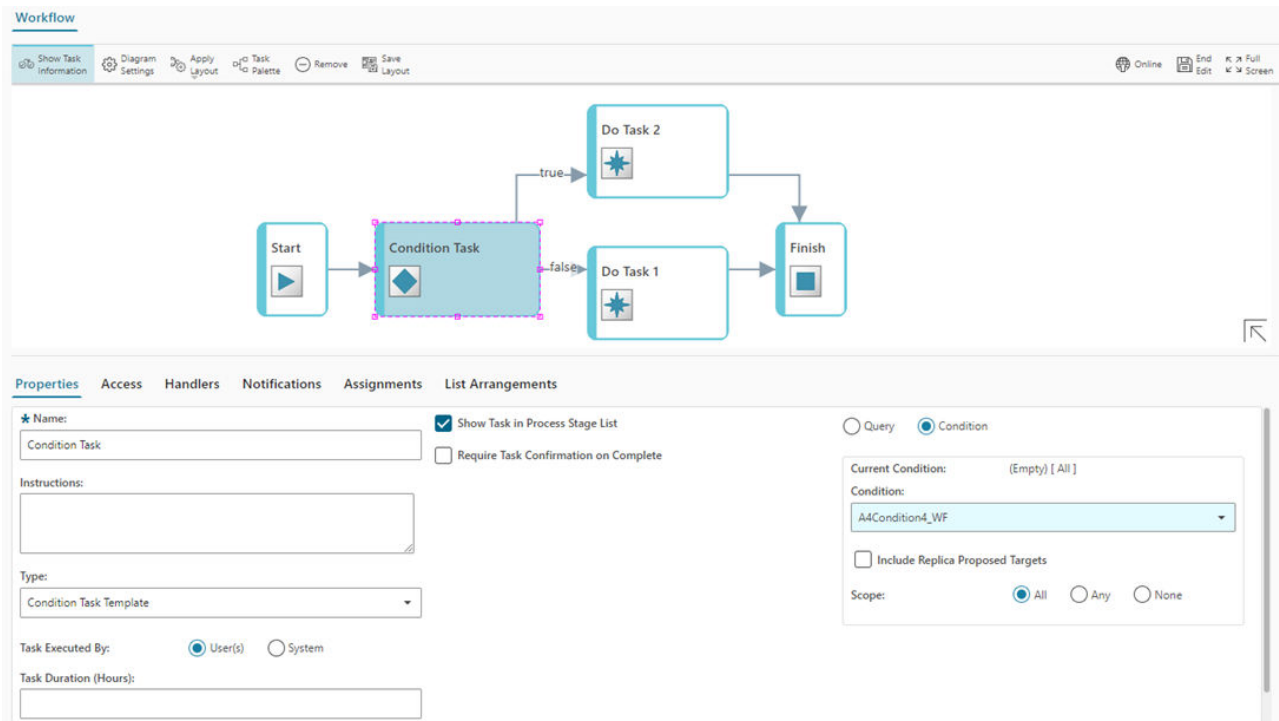
A pre-existing BMIDE condition must be available before applying it to a Condition task. There are predefined generic conditions available in the BMIDE that you can reuse, or you can also create custom conditions if needed using the new condition wizard.

Procedure

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

- Click the **Condition** drop-down list and select a condition to apply to the task, or enter a search string to locate one.
- Click the **Include Replica Proposed Targets** check box to include targets for the remote workflow task in the search.
- 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.



Add condition queries to a Condition task

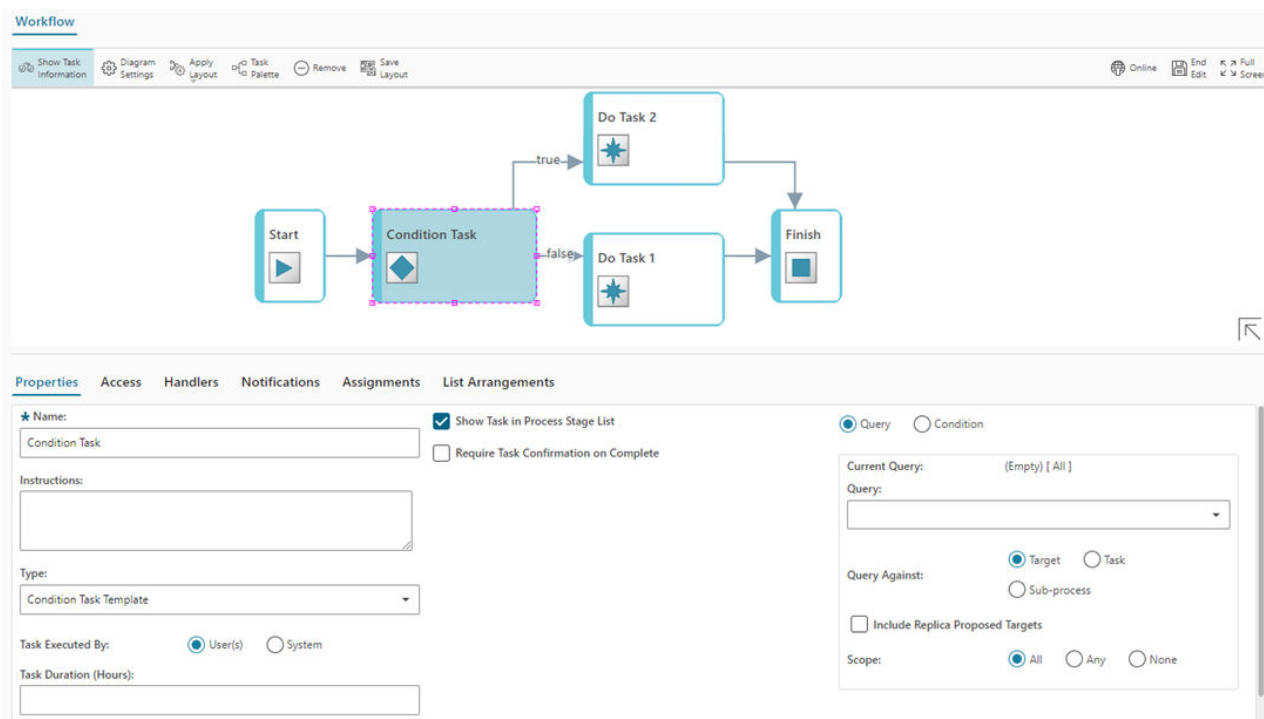
Condition tasks have optional properties for task behavior. Condition queries control the specific functions of a **Condition** task. Once applied, conditions check for specific parameters that you set. These conditions work in conjunction with the **Success** path from the **Condition** task to its task or target.

Procedure

- In the **Properties** tab of the **Condition** task, click **Query**.
- Select a query from the **Query** list. Queries are created in the rich client.
Your organization may have their own configured queries in this list.
- 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.



Configure how tasks display in a workflow template

Set the visibility of tasks


Workflow Designer provides the option to make tasks visible or non-visible to end users in a workflow template. Administrators can display a user-interactive task in a workflow, or hide tasks from users that are non-interactive, without specified assignments, or are automatically completed by a handler.

Filtering the visibility of tasks is convenient if you have a very large template with numerous tasks. Hiding non-interactive (or automated) tasks can tidy up a template's appearance and create less graphical clutter for the end user, as task nodes are minimized on the template's graphical interface and display only as icons. This task filtering then applies to various task lists, such as when submitting an object to a workflow, identifying task assignments or viewing upcoming tasks in the Inbox, and when viewing a process assignment list (PAL).

Note:

The Review, Acknowledge, and Route tasks always display in a template and their visibility settings cannot be configured.

Procedure

1. Open the template that contains the task(s) 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 specific task to modify.
4. Click the **Properties** tab.
5. Under **Task Executed By**:
 - Click **System** to specify that it is a non-interactive, automated task that is not visible to the user.
 - Click **User** to specify that the task is interactive and will be visible to the user.

Workflow

Show Task Information | Diagram Settings | Apply Layout | Task Palette | Remove | Save Layout | Online | End Edit | Full Screen

Properties | Access | Handlers | Notifications | Assignments | List Arrangements

Type: Add Status Task Template

Task Executed By: ☐ User(s) ☒ System

Task Duration (Hours):

Release Status: Translation Submitted

Once online, the template reflects the task visibility updates in the following areas:

- When submitting a workflow, in the **Submit to Workflow** panel
- In the Inbox:
 - The Assignment tab
 - The Upcoming Tasks table on the Workflow tab
 - The graphical view on the Workflow tab; non-interactive tasks only display their **icon**.



- In an Assignment List

Set the task order


Workflow Designer provides the option to configure the task order in a workflow template that is then reflected in other task lists.

For example, if you **add a new task to a template** from the **Task Palette**, that task is automatically displayed at the bottom of any workflow task list unless you manually change its order in the Workflow Designer.



Note:

The reordering of child tasks in the Review, Acknowledge, and Route tasks cannot be configured.

Procedure

1. Open the template that contains the task(s) you want to modify, and click **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 **Properties** tab.
4. Click the **List Assignments** tab.

Tasks are listed in the order that they are in the template.

5. Change the task order by selecting a task row and clicking **Move Up**  or **Move Down**  (on top of the table) to move the task up or down, respectively.

Or, drag and drop the tasks to the correct order in the table.

This will represent the order that the tasks will execute in the template.

Once online, the updated task order is reflected in the following task lists:

- When submitting a workflow, in the **Submit to Workflow** panel
- In an Assignment List
- In the Inbox:
 - The Assignment tab
 - The Upcoming Tasks table on the Workflow tab

Check for errors with a Validate task

Find error codes

Most workflow errors are displayed within the Enterprise Process Modeling (EPM) module. All error codes are documented in the *Integration Toolkit Function Reference*. Here, error codes are grouped by module. For example, Application Encapsulation (AE) errors are listed within the AE module, Appearances errors are listed within the Appearances module, and so forth.

Although using workflow (EPM) error codes with the **Validate** task may be the most common usage, the task works with any error code. You can add error codes from any module, or custom error codes, to the **Results List**.

This procedure describes how to display a list of error messages.

Procedure

1. Open the *ITK Function Reference* on Support Center, under **Teamcenter Documentation, Teamcenter x Developer References** (where x is the release).
2. At the top of the page, select the **Modules** header.
3. In the **Modules** page, scroll down to the appropriate module.

For example, to see all Enterprise Process Modeling (EPM) errors, which contain the majority of workflow errors, scroll to **Enterprise Process Modeling (EPM)** and click the link.

4. Click **Errors**.

The error page displays all errors for that module. Error numbers are defined as *module base value + error code*.

For example, the **EPM_internal_error** error has an error code of **EMH_EPM_error_base + 1**.

Add error codes

In the context of the **Validate** task, a *workflow error* means any error generated by a workflow handler. After drawing a failure path between the **Validate** task and a successor task, you must specify how you want the failure path to respond to workflow errors. You configure a failure path by **pairing** a workflow handler and an error code.

The failure path can be configured to activate when:

- Any error occurs by selecting **All Errors**.

This option automatically configures the failure path to activate upon any error. No additional steps are required.

- *Specific errors occur by selecting **Selected Errors**, as shown in the following procedure.*

Procedure

1. Click the path you want to configure as a failure path.

The **Properties** for the path display.

2. Under **Results**, click **Selected Errors** to specify which error codes you want the Validate task to check.

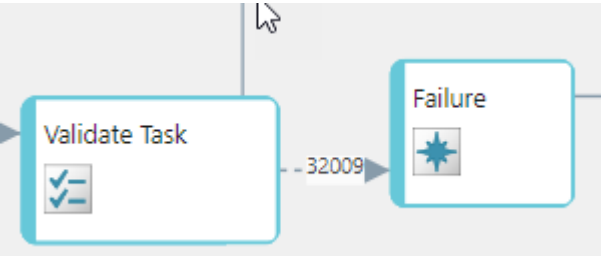
The **Add Error Code** text box displays.

3. In the **Add Error Code** box, type an EPM error code. For example, type **32009** (RES_OBJECT_IS_RESERVED) to ensure the failure path is followed whenever a target is not checked in.
4. Click **Add** to add this error to the **Results List**.

The screenshot displays the Workflow Designer interface. The top section, titled "Workflow", shows a diagram with the following steps: Start → Do Task → Validate Task. From the "Validate Task" step, there are two paths: a solid line leading to a "Success" box and a dashed line leading to a "Failure" box. Both paths then lead to an "Or Task" box, which finally leads to a "Finish" box. The bottom section, titled "Properties", shows the configuration for the selected "Failure" path. It includes a "Predecessor:" field with "Validate Task" and a "Successor:" field with "Failure". Under the "Results:" section, the "Selected Errors" radio button is selected. The "Add Error Code:" text box contains the value "32009", and the "Add" button is highlighted with a mouse cursor. Below this, the "Results List:" dropdown menu is set to "ANY".

5. Continue adding errors to the **Results List** until you have specified all the errors you want to cause the workflow to follow the failure path.

The specified errors display on the failure path.



Validate task behavior

The **Validate** task's behavior depends upon how its failure path is configured and what errors are received.

Failure criteria you specified	Error thrown (if any)	Task behavior
Fail if any error	Any error	Failure path is followed.
Fail if error on error list occurs	Error on error list	Failure path is followed.
Fail if error on error list occurs	Error <i>not</i> on error list	Workflow process halts. Task remains in Started state and an error appears.
No failure path configured	Any error	Workflow process stops. Task remains in Started state and an error appears.
Regardless of whether failure path was configured, and whether errors occurred	No errors occur	Success path followed. If no success path was configured, workflow process stops.

7. Using an assignment matrix

What is an assignment matrix?

An *assignment matrix* is a responsibility charting tool that identifies responsibilities for everyone on the team, including specific duties and how people participate in the project. By constructing an assignment matrix, a project manager can make sure that the team members maintain their specific roles and assignments. This can help to avoid confusion as the project progresses.

You can use an assignment matrix to automate the assignments of workflow participants, eliminating the need for end users to make decisions about task participation in a workflow. The assignment matrix creates an automated mapping between the task activities and the end users for easier and faster workflow execution. You can predefine assignees for various roles (or **Assignment Types**) based on identified property values (or **Decisive Properties**) for the target object.

For example, the assignment matrix named **ECN Assignment Matrix** in the following graphic would be used if:

The **Target Type** of **Change Notice Revision** (1) is included in a workflow.

The **Decisive Properties** (2) for the target include **Description=New**, **Disposition=Approved**, and **Closure=On Hold**.

If those criteria are met and the **AMX handler** is configured on the task to pick up assignments from the assignment matrix, then the assignment matrix automatically assigns the user **Tester, Tammie** (from the **Engineering** group/**Designer** role) as the **Assignee**, and **Manager, Mikey** and **Analyst, Amy** each as a **Reviewer** (3) for the task in all affected workflows.

ECN Assignment Matrix
Name: ECN Assignment Matrix | Description: | Release Status: | Type: Assignment Matrix

PROPERTIES

Name: ECN Assignment Matrix
Description: | Owner: Writer, Wilma (wilma)
Group ID: Engineering
Creation Date: 02-May-2023
Checked-Out: |
Checked-Out By: |

Application Context: General
Target Type: Change Notice Revision (1)
Assignment Matrix Definition: General-Change Notice Revision
Decisive Properties: Description, Disposition, Closure

ENTRIES

Decisive Properties	Assignment Type	Assignees
Description=New, Disposition=Approved, Closure=On Hold (2)	Assignee	Engineering/Designer/Tester, Tammie (tammie) (3)
Description=New, Disposition=Approved, Closure=On Hold	Reviewer	Engineering/Designer/Manager, Mikey (mikey), Engineering/Designer/Analyst, Amy (amy)

DECISIVE PROPERTIES

Property Name	Property Value
Closure	On Hold
Disposition	New
Disposition	Approved

Note:

An assignment matrix is not the same as a process assignment list (PAL). A PAL uses a human decision when determining the participants for an object in a workflow. An assignment matrix is configured before the workflow runs, and assignments are automated based on the data submitted to the workflow.

The assignment matrix process

The following is a high-level overview of the steps involved when creating and using an assignment matrix.

1. Administrative tasks

Tasks to be handled by an administrator before using an assignment matrix include:

- **Modify the style sheet**

Assignment matrices are not available out of the box for any business object. You must modify the object style sheet to include the **Assignment Matrices** tab when viewing an object that contains an assignment matrix.

- **Add additional LOV selections when creating an assignment matrix**

The **Application Context** and **Assignment Type** drop-down menus are used during the assignment matrix creation process. Additional options for both can be added using BMIDE.

2. Create an assignment matrix definition

Access the **Assignment Matrix Definitions** tile on the home page to begin. An assignment matrix definition includes the details that will apply to an assignment matrix.

3. Create and populate an assignment matrix based on the assignment matrix definition

Once created, the assignment matrix uses the pre-existing definition to determine user participation once it is applied to a specific business object.

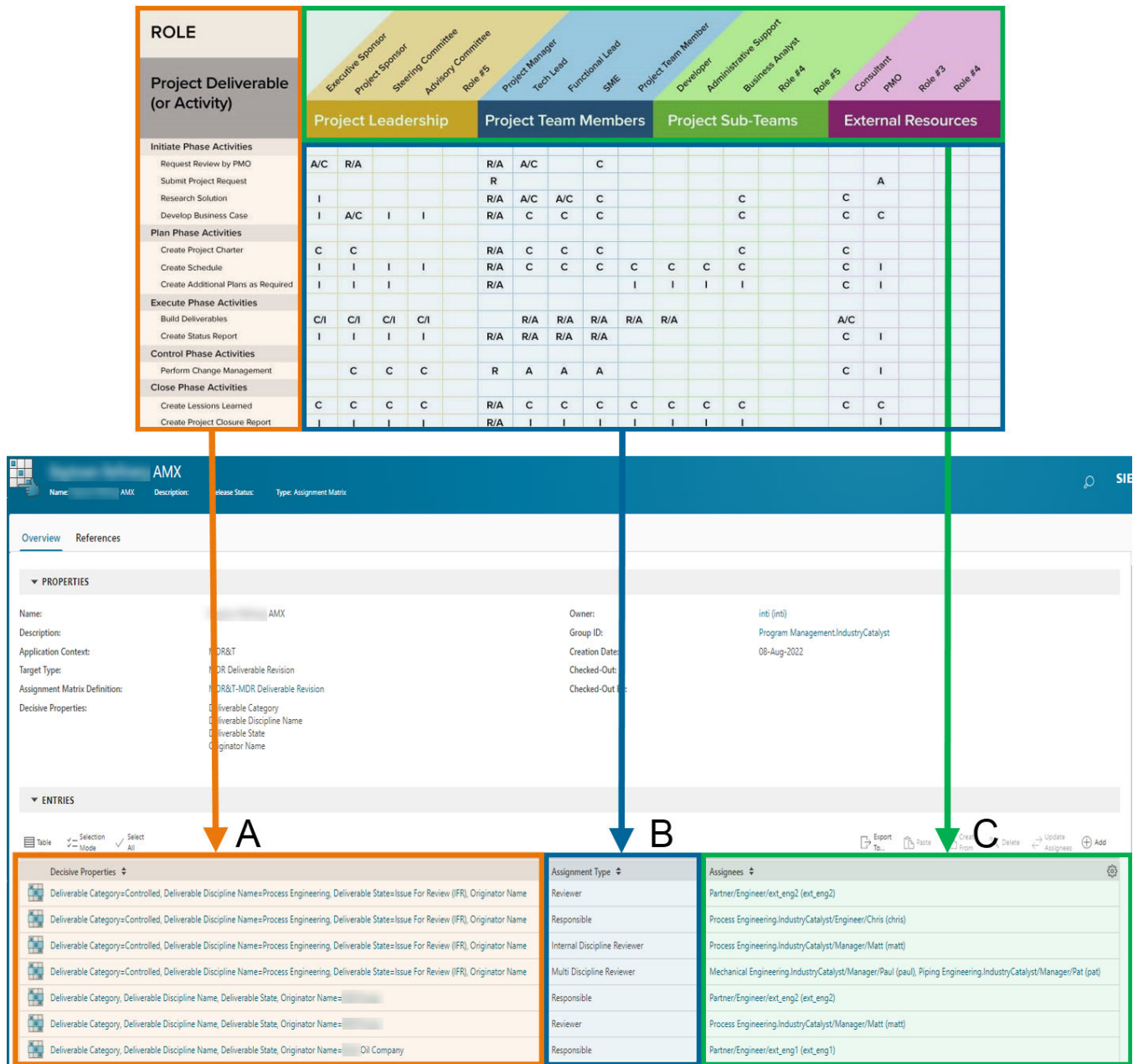
4. Add and modify the AMX handler

A task must include the AMX handler to use an assignment matrix. The AMX handler sets participants based on target properties (which are specified in the assignment matrix definitions). The handler reads various fields from the assignment matrix and identifies an assignment type specified in its configuration. Then the workflow is automatically routed accordingly. Use this handler to set dynamic participants on the target object as well.

Compare a RASIC table to an assignment matrix

A RASIC chart (responsible, approving, supporting, informed, and consulting) is often used in a project to ensure the clarity of roles and to ensure that responsibilities are assigned. It can save time and lessen workload by defining and communicating the roles at the beginning of the project. A RASIC chart works similarly to an assignment matrix in the way that task assignments are identified and automated before a workflow executes.

The following image shows how a RASIC chart compares to an assignment matrix.



The rows in the RASIC chart that represent deliverables are mapped to column A in the assignment matrix. These deliverables are identified in Active Workspace by their specific property values (or decisive properties).

The cell values in the RASIC chart are mapped to the **Assignment Type** in the assignment matrix (B).

The columns in the RASIC chart that represents roles are mapped to the **Assignees** column in the assignment matrix (C).

For example, the first entry in the **Assignees** column, or column C in the image, identifies that the user **ext_eng2** (from the **Partner** group with the **Engineer** role) is assigned as the **Reviewer** for all the deliverables that contain these decisive properties:

- **Deliverable Category** is **Controlled**
- **Deliverable Discipline Name** is **Process Engineering**
- **Deliverable State** is **Issue For Review**

Administrative tasks

Using an assignment matrix in a workflow requires the following administrative tasks.

Modify the object style sheet

You can control the layout for certain portions of the declarative interface by using XML rendering templates (XRT), also called *style sheets*. These XML files are stored in the Teamcenter database and are read as needed. For this reason, changes made to these rendering templates are reflected in the UI without the need to build or deploy an application file. To view the assignment matrix details for an object, you must modify the object's style sheet to include the **Assignment Matrices** tab.

1. From the **XRT editor**, open the style sheet for the object that you want to contain the assignment matrix.
2. Enter the following line:

```
<inject type="dataset" src="Amx1AssignmentMatrixesPage"/>
```

The **Assignment Matrices** tab is displayed.



Add additional LOV selections for the assignment matrix

- The **Application Context** drop-down menu identifies the environments that the assignment matrix will be used in.

Create Assignment Matrix Definition ✕

* Application Context:
Required ▼

* Target Type:
Required ▼

* Decisive Properties:
Required ▼

General is provided out-of-the-box, but additional selections can be added in BMIDE by modifying the **Amx0ApplicationContext** LOV.

- The **Assignment Type** drop-down menu is used when adding entries to the assignment matrix once it has been created.

Add ✕ Close Pin Panel

▼ DECISIVE PROPERTIES

Property Name	Property Value	⚙️
Description		

* Assignment Type:
Required ▼

▼ ASSIGNEES

⊕ Add

Additional selections can be added in BMIDE by modifying the **Amx0AssignmentType** LOV.

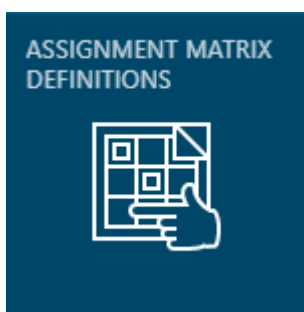
Create an assignment matrix definition


An assignment matrix definition includes the details you would like to apply to an assignment matrix, based on the target object type. Think of it as a container that holds all the rules for an object to proceed through a workflow without user interaction. The assignment matrix uses this definition to determine user participation once it is applied to a specific object.

There are no permissions needed to create an assignment matrix definition, so any user can create one.

Procedure

1. From the **Home** page, click the **Assignment Matrix Definitions** tile.



2. Go to **More Commands** ... > **New** ✱ > **Create Assignment Matrix Definition** .
3. Select an existing **Application Context** from the drop-down.

The **Application Context** identifies the environment that the assignment matrix will be used in. This must be a unique value.

4. Select the **Target Type** that will contain the assignment matrix. This must be a unique value.

Note:

An assignment matrix definition defined on a parent class can be used for an instance corresponding to a sub-class. For example, an ItemRevision-based assignment matrix definition can be used on a subclass of ItemRevision class.

5. Select the **Decisive Properties** for the target from the drop-down menu.

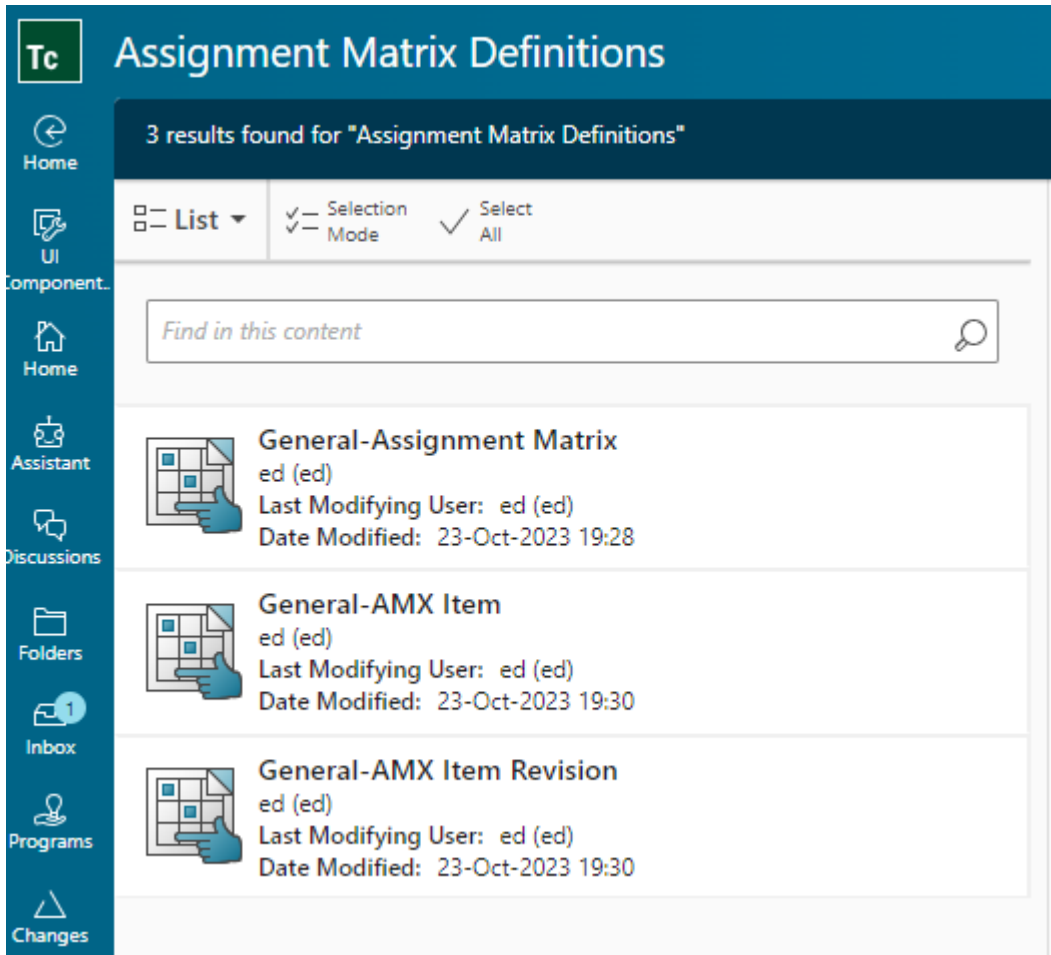
Decisive properties are values that describe how to identify the selected target so that the assignment matrix can assign participants.

Note:

Runtime properties are not supported for decisive properties.

6. Click **Create**.

The new definition is listed on the **Assignment Matrix Definition** panel. Once selected, the properties are displayed for the definition.



Create and populate an assignment matrix

The assignment matrix contains the assignment details for the target object as defined in the assignment matrix definition. Once the assignment matrix is created, you can populate it with specific properties for the object so that the correct assignees are identified for it during a workflow.

Prerequisites

An assignment matrix definition must be created.

Procedure

1. Create the assignment matrix.

- a. Open the target object to that you want to contain the assignment matrix.
- b. Click the **Assignment Matrices** tab.

Note:

The object's style sheet must be modified for this tab to appear.

- c. Click **Add** ⊕ in the top-right portion of the **Assignment Matrices** table.
- d. Enter a **Name** for the assignment matrix.
- e. Select the **Application Context** from the drop-down menu.
- f. Select the **Target Type**.

If an assignment matrix definition is identified for the target, the **Target Type** is automatically displayed.

- g. Enter a **Description** for the matrix.

The **Assignment Matrix Definition** is automatically selected based on the data entered, and its **Decisive Properties** are displayed.

- h. Click **Open on Add** to open the assignment matrix immediately.
- i. Click **Add**, and the new assignment matrix appears.

2. Associate decisive properties with assignees.

- a. Under **Entries**, click **Add** ⊕ above the **Decisive Properties** table.
- b. The **Property Name** is populated from the assignment matrix definition. For each property, enter a **Property Value** to be associated with any assignees.
- c. Select an **Assignment Type** from the drop-down menu.
- d. Under **Assignees**, click **Add** ⊕.
- e. In the **Add** dialog box, type the name of the user to whom you want to assign the task that contains the object.

- f. Press **Enter** or click **Search** to see the results.

Tip:

You can enter a partial name or wildcard characters.

- g. (Optional) Click **Filter**  for additional search filters.

The default filters are **Type**, **Group**, or **Role**. Select one of the following options to determine which users appear in the list.


- Under **Type**, specify either a **Group Member** or **Resource Pool**.
- Specify group members by **Group** or **Role**. Enter a name in the specific search box to filter.

Note:

You can also display the filters and increase the size of the **Add** panel by dragging the left side of the panel to the left.

- h. Click **Add** to include any assignee(s).
- i. Click **Add** to add the assignment matrix entry.

Continue adding entries to the assignment matrix as necessary. Decisive properties, their respective values, and the assignment type must be unique for each assignment matrix row.

Click **Table Settings**  > **Wrap Text** to view all of the assignees in a row.

Note:

Decisive property values can be blank. If a value for a certain decisive property is blank, that property value will be marked as matched, regardless of any value present on the target object. This process functions similarly to using match all and it includes null and non-null property values.

Update an assignment matrix

Once an assignment matrix entry is created, click **Edit**  > **Start Edit**  to modify it in the following ways.

- **Save as**

Creates a new entry based on the selected entry.

- **Delete**

Deletes the selected entry.

- **Update Assignees**

Update any assignees for the selected entry.

- **Export To**

Exports the selected entry to an Excel-compatible file format.


Add and modify the AMX handler

The **AMX-auto-assign-task-assignees-from-assignment-matrix** handler must be added to a task to get assignees from an assignment matrix. This handler automatically creates assignments during the workflow process based on the entries of the related assignment matrix. If a workflow contains an AMX handler on a task matching the target object's decisive properties (as defined in the assignment matrix rows), those assignments from the matrix are made automatically.

The handler uses arguments to identify various assignment combinations for an object. Arguments can consider such things as assignees being merged or overridden, situations when there is only a partial match for an object, the use of dynamic participants, and so on. Any workflow template can add this handler to use an assignment matrix.

See the handler documentation for detailed examples.

Procedure

1. Open the template that contains the task that you want to contain an assignment matrix, and click **Edit**  > **Start Edit**.
2. Click the task to include the assignment matrix.
3. Click the **Handlers** tab.

In the left pane, the table lists any handlers assigned to the selected task. Handlers are listed under their specific trigger group in order to represent the running order of action and rule handlers. To see any handlers currently assigned under a specific trigger, click expand ▼.

4. Select a trigger group to add the handler to.
5. In the **New Handler** panel on the right, click **Action** for the handler type.
6. All the applicable handlers for the trigger group are available in the drop-down list. The handler list is not case-sensitive and it may contain custom handlers as well as handlers that are included with other solutions.

Select the **AMX-auto-assign-task-assignees-from-assignment-matrix** handler.

The mandatory arguments are displayed.

The screenshot shows a configuration window for a handler. At the top, 'Handler Type' has two radio buttons: 'Action' (selected) and 'Rule'. Below this, the 'Name' is set to 'AMX-auto-assign-task-assignees-from-assignment-matrix' with a dropdown arrow and a 'Help' icon. A '+ Add Arguments' button is visible. The 'Argument' section lists three arguments: '-target_type', '-application_context', and '-assignment_type'. Each argument has a corresponding 'Value(s)' field. For '-target_type', the field contains 'Click to add a value'. For '-application_context' and '-assignment_type', the fields contain 'Type in required value(s)'. An 'Add' button is at the bottom right.

7. Enter the corresponding values in the **Value(s)** field.

Note:

Argument descriptions are listed in the handler documentation.

8. To add additional arguments, click **Add Arguments** ⊕ and select the new argument from the drop-down list.
9. Enter any 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 while avoiding runtime errors.

10. Click **Add**.

After completing this step, an assignee will be automatically assigned to the task if the corresponding assignment matrix matches these specific properties.

Note:

Multiple **AMX-auto-assign-task-assignees-from-assignment-matrix** handler entries can be added for a single task definition.

Expand the trigger group to view the new AMX handler in the table. Once added, you can highlight the handler and click **Cut** ✂, **Copy** 📄, or **Delete** ☹. To make any additional changes to the argument, highlight the row and edit the handler in the **Information** panel on the right.