

Sub Workflow Activity Overview

Last Modified on 06/08/2021 10:20 am EDT

V8.2 and later

Context

Use the Sub Workflow Activity to kick off another workflow from within the parent workflow. The sub workflow is another workflow that is defined in Cora SeSequence. Sub workflows are useful to promote encapsulation, such as when the same business process is used in multiple scenarios, or to break down processes to smaller modules. You can also use sub workflows as a type of "black box", such as when a specific part of the process needs to be hidden from certain users.

Sub workflows are executed as a separate workflow instance, like any other workflow. Sub workflow execution can be synchronous or asynchronous, which means that depending on the business need, the sub workflow's execution can block the parent workflow's execution until it is complete, or it can allow the parent workflow to continue without requiring that the sub workflow is completed. To control this functionality, when you design a sub workflow, set the **Blocking** and **Show in tree** properties of the sub workflow as necessary.

Use Cases

- **Encapsulation:** reuse a workflow that assigns a technician to a service request by starting the workflow from two different workflows. For example, an equipment installation workflow and a service request workflow.
- **Black Box:** create a sub workflow that handles special budget requests that require discrete processing and is not visible from the standard budget request process.

Delete and Rollback Behavior

You can determine what happens to an open sub workflow instance if its main workflow instance is deleted or rolled back by a Flowtime user (this user must have full control permissions). You can do this by setting the **Delete** and **Rollback** properties in the **Behavior** section of the sub workflow's Properties pane. These two properties can be set to the values: **None**, **Cascade** or **Restrict**.

If a sub-workflow exists, but has not been reached in the workflow, these options are not relevant.

These settings have the following affect upon deletion/rollback of a workflow instance if an open sub workflow instance exists:

Setting	Main workflow instance	Sub workflow instance
None	Is deleted / is rolled back	Is not deleted / is not rolled back
Cascade	Is deleted / is rolled back	Is deleted / is rolled back
Restrict	Is not deleted / is not rolled back *an error message is displayed to the user.	Is not deleted / is not rolled back

For example, if the Rollback property is set to None, then if a user rolls back a main workflow instance 'A',

with an open sub workflow instance 'B', then 'A' will be rolled back, but 'B' will not.

Wizard Tips

- The **Blocking** property indicates a synchronous workflow execution, *True*=synchronous, *False*=asynchronous.
- The **Show in tree** property indicates whether or not the sub workflow and its activities are visible and accessible from the Flowtime tree of the parent workflow (main workflow).
- Select the **One way** property check box to indicate that the parent workflow will not resume when the sub workflow finishes. This option is only relevant when you use non-blocking mode.
- To access the wizard, right-click the Activity and select **Edit**, and complete the wizard.
- Connect the And Activity to at least one source activity.
- Only map fields with matching types.
- You can only insert sub workflow fields into fields of the parent workflow activities that were executed before the sub workflow activity in the parent workflow.
- You can edit the sub workflow directly by right-clicking the activity.

Sub Workflow Activity Transformation Definition Step 1

Sub Workflow Name: ...

Select a target workflow and build the transformation definition

Main Workflow Fields	Transformation Table	Sub Workflow Fields						
Double-click an item to select		Double-click an item to select						
<ul style="list-style-type: none">▶ <> (metadata)▶ Dept Approval▶ End▶ Notify Manager▶ OpenAccount<ul style="list-style-type: none">▶ <> (metadata)▶ OpenAccount<ul style="list-style-type: none">fldIdfldIWfidfldIActIdfldAIIdfldMasterIWfidCompanyBranchAddress▶ Start	<table border="1"><tbody><tr><td>{OpenAccount}.Query("Open. ...</td><td>wf.Variables["Credit"]</td><td>⊗</td></tr><tr><td>{OpenAccount}.Query("Open. ...</td><td>wf.Variables["Status"]</td><td>⊗</td></tr></tbody></table>	{OpenAccount}.Query("Open. ...	wf.Variables["Credit"]	⊗	{OpenAccount}.Query("Open. ...	wf.Variables["Status"]	⊗	<ul style="list-style-type: none">▶ <> (metadata)▶ Variables<ul style="list-style-type: none">CreditStatus
{OpenAccount}.Query("Open. ...	wf.Variables["Credit"]	⊗						
{OpenAccount}.Query("Open. ...	wf.Variables["Status"]	⊗						
	Add new transformation							

Back Next Cancel

Activity Tips

- Add the sub workflow as an activity inside the parent workflow.
- Sub workflows are executed as a separate workflow instance, like any other workflow. Sub workflow execution can be synchronous or asynchronous, which means that depending on the business need, the sub workflow's execution can block the parent workflow's execution until it is complete, or it can allow the parent workflow to continue without requiring that the sub workflow is completed.
- The user that creates the sub workflow activity in the parent workflow must have Read & Write permissions for the first activity of the sub workflow. The user that completes the sub workflow must have Read & Write permissions for the next activity in the parent workflow.
- Input parameters enable you to pass values from the parent workflow to the sub workflow as defined in the *Parent to Son Transformation* table, which is a property of the sub workflow activity.
 - Assign values to the variables of the sub workflow.

- Assign values to the fields in the first form of the sub workflow. This option causes the first form to submit automatically and redirect to the next activity.
- Output (returned) parameters enable you to pass values from the sub workflow to the parent workflow as defined in the *Son to Parent Transformation* table, which is a property of the sub workflow activity. These values are returned when the sub workflow instance finishes execution, whether it is synchronous or asynchronous.
- Set up the **Run As** property to define the user who needs to execute the sub workflow.

V7.x

Context

Use the Sub Workflow Activity to kick off another workflow from within the parent workflow. The sub workflow is another workflow that is defined in Cora SeQUENCE. Sub workflows are useful to promote encapsulation, such as when the same business process is used in multiple scenarios, or to break down processes to smaller modules. You can also use sub workflows as a type of "black box", such as when a specific part of the process needs to be hidden from certain users.

Sub workflows are executed as a separate workflow instance, like any other workflow. Sub workflow execution can be synchronous or asynchronous, which means that depending on the business need, the sub workflow's execution can block the parent workflow's execution until it is complete, or it can allow the parent workflow to continue without requiring that the sub workflow is completed. To control this functionality, when you design a sub workflow, set the **Blocking** and **Show in tree** properties of the sub workflow as necessary.

Use Cases

- **Encapsulation:** reuse a workflow that assigns a technician to a service request by starting the workflow from two different workflows. For example, an equipment installation workflow and a service request workflow.
- **Black Box:** create a sub workflow that handles special budget requests that require discrete processing and is not visible from the standard budget request process.

Delete and Rollback Behavior

You can determine what happens to an open sub workflow instance if its main workflow instance is deleted or rolled back by a Flowtime user (this user must have full control permissions). You can do this by setting the **Delete** and **Rollback** properties in the **Behavior** section of the sub workflow's Properties pane. These two properties can be set to the values: **None**, **Cascade** or **Restrict**.

If a sub-workflow exists, but has not been reached in the workflow, these options are not relevant.

These settings have the following affect upon deletion/rollback of a workflow instance if an open sub workflow instance exists:

Setting	Main workflow instance	Sub workflow instance
None	Is deleted / is rolled back	Is not deleted / is not rolled back
Cascade	Is deleted / is rolled back	Is deleted / is rolled back

Restrict	Is not deleted / is not rolled back *an error message is displayed to the user.	Is not deleted / is not rolled back
-----------------	---	-------------------------------------

For example, if the Rollback property is set to None, then if a user rolls back a main workflow instance 'A', with an open sub workflow instance 'B', then 'A' will be rolled back, but 'B' will not.

Wizard Tips

- The **Blocking** property indicates a synchronous workflow execution, *True*=synchronous, *False*=asynchronous.
- The **Show in tree** property indicates whether or not the sub workflow and its activities are visible and accessible from the Flowtime tree of the parent workflow (main workflow).
- Select the **One way** property check box to indicate that the parent workflow will not resume when the sub workflow finishes. This option is only relevant when you use non-blocking mode.
- To access the wizard, right-click the Activity and select **Edit**, and complete the wizard.
- Connect the And Activity to at least one source activity.
- Only map fields with matching types.
- You can only insert sub workflow fields into fields of the parent workflow activities that were executed before the sub workflow activity in the parent workflow.

Activity Tips

- Add the sub workflow as an activity inside the parent workflow.
- Sub workflows are executed as a separate workflow instance, like any other workflow. Sub workflow execution can be synchronous or asynchronous, which means that depending on the business need, the sub workflow's execution can block the parent workflow's execution until it is complete, or it can allow the parent workflow to continue without requiring that the sub workflow is completed.
- The user that create the sub workflow activity in the parent workflow must have Read & Write permissions for the first activity of the sub workflow. The user that completes the sub workflow must have Read & Write permissions for the next activity in the parent workflow.
- Input parameters enable you to pass values from the parent workflow to the sub workflow as defined in the *Parent to Son Transformation* table, which is a property of the sub workflow activity.
 - Assign values to the variables of the sub workflow.
 - Assign values to the fields in the first form of the sub workflow. This option causes the first form to submit automatically and redirect to the next activity.
- Output (returned) parameters enable you to pass values from the sub workflow to the parent workflow as defined in the *Son to Parent Transformation* table, which is a property of the sub workflow activity. These values are returned when the sub workflow instance finishes execution, whether it is synchronous or asynchronous.