

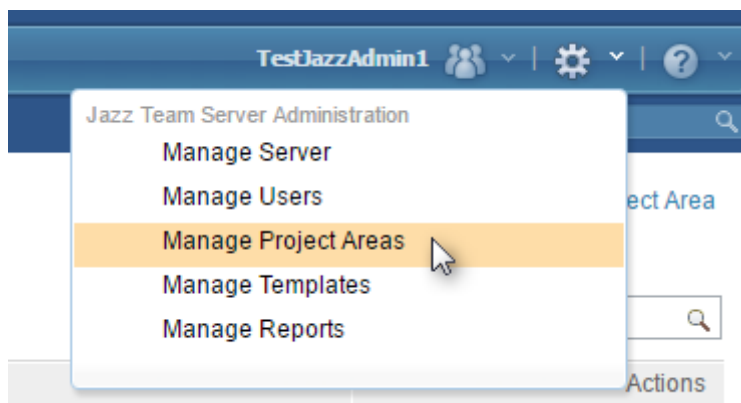
## Plugin Configuration in RTC

### Configure Stages Workflow Engine Precondition & Follow-up action

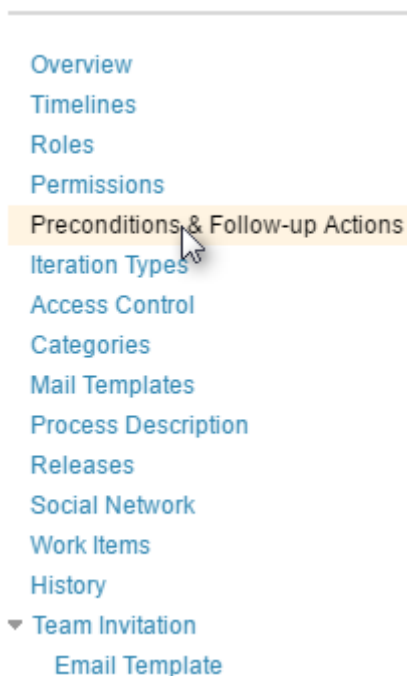
The Plug-in has to be activated for a specific operation in the project area configuration. For each operation, several preconditions and follow-up actions can be assigned.

The precondition & follow-up action of the Stages plug-in can be added per operation. If you want to have the plug-in activated for several operations, you have to repeat the following steps for each supported operation.

Log in to Jazz Team Server as a user with administrative privileges for the concerning project area. Select the gear icon and select Manage Project Areas.



Select the project area you want to prepare for the Stages plug-in usage. On the administration page of the selected project area click on Preconditions & Follow-up Actions.



On the Precondition & Follow-Up Actions management page look for the Action Save Work Item. Hover over this action with your mouse and select the edit button.

Preconditions & Follow-up Actions Configure

Previous | 1 - 2 of 2 | Next

What	Who	When	Summary	Actions
Generate Team Invitation	Everyone (default)	Always	1 follow-up action	
Save Work Item	Everyone (default)	Always	1 precondition, 0 follow-up actions	

On the action configuration window click on Add.

**Save Work Item**

Preconditions & Follow-up Actions: Add Choose a precondition or follow-up action from the left to see more details.

Required Attributes F...

Final (ignore customization of this operation in child areas)

Who: Everyone (default) Change...

When: Always Change...

Select the Stages Workflow Engine under preconditions and click OK.

**Add Precondition or Follow-up Action**

Choose a precondition or follow-up action to configure.

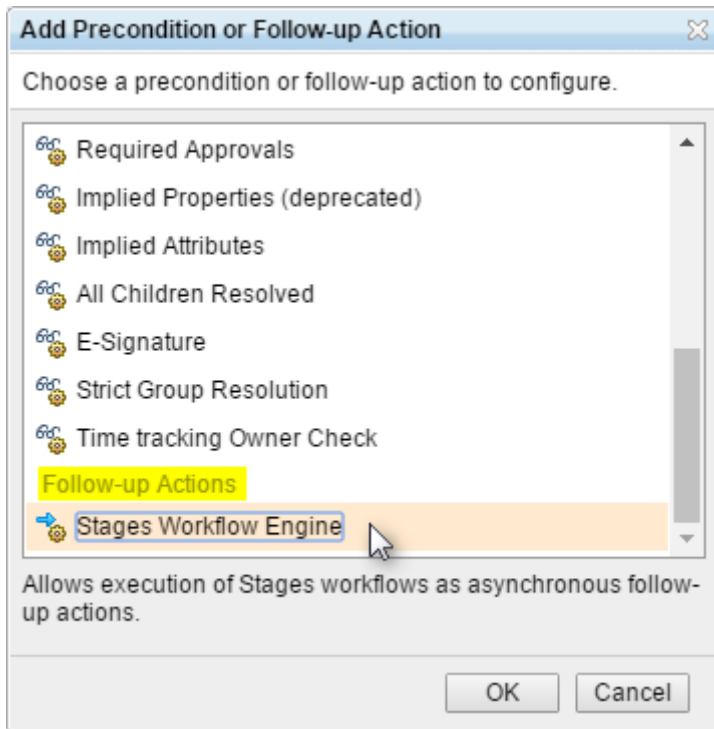
**Preconditions**

- Prevent Editing
- Stages Workflow Engine
- Require Disinterested Approvers
- Required Attributes For Type and State
- Required Attributes For Condition
- Read-Only Attributes For Type and State
- Read-Only Attributes For Condition
- Attribute Validation

Allows execution of Stages workflows as synchronous preconditions.

OK    Cancel

Back on the action configuration window click on Add again and this time scroll to the follow-up actions at the bottom of the Add Precondition or Follow-up Action window. Select the Stages Workflow Engine again and click OK.



To verify this steps you can check the preconditions and follow-up actions in the overview.

Preconditions & Follow-up Actions ? Configure

Previous | 1 - 2 of 2 | Next

What	Who	When	Summary	Actions
Generate Team Invitation	Everyone (default)	Always	1 follow-up action	
Save Work Item	Everyone (default)	Always	2 preconditions, 1 follow-up action	

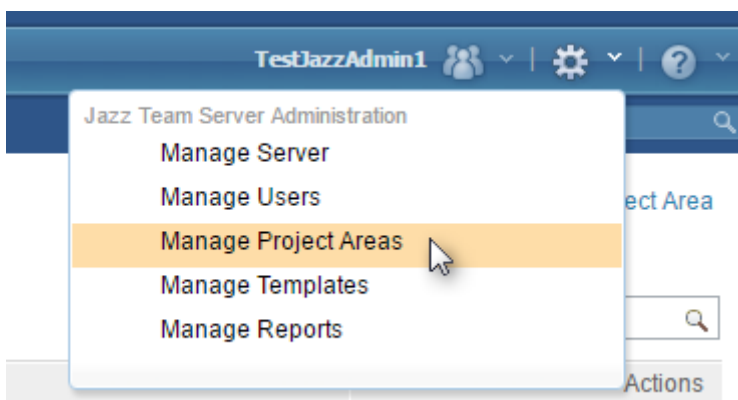
After these steps your project area is ready to use the Stages plug-in for the Jazz Team Server.

### Integrate Stages Links in Editor Representations

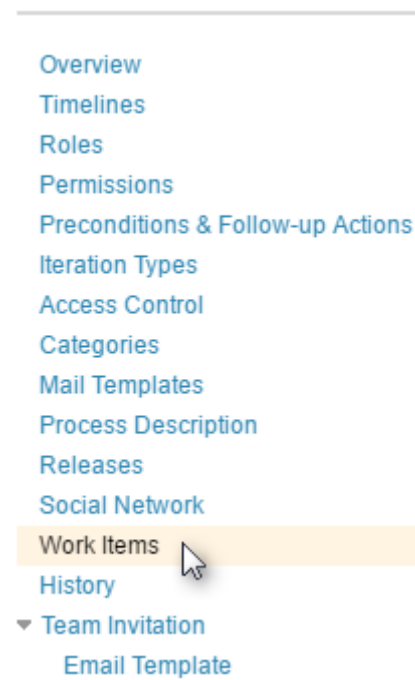
The Stages plug-in for the Jazz Team Server stores links back to the corresponding Stages process element in a hidden attribute of each work item created by the plug-in. This attribute is created automatically and is initially not visible to the user (e.g. in the editor page of a work item of the web application).

If you want to make these links visible for users of the work item editor in the web application follow this steps:

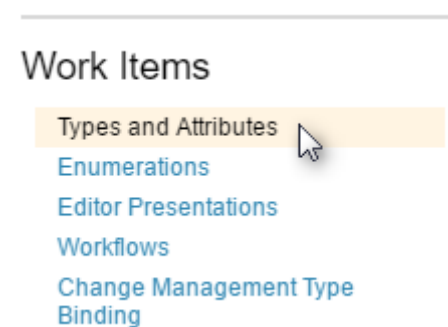
Log in to Jazz Team Server as a user with administrative privileges for the concerning project area. Select the gear icon and select Manage Project Areas.



Select the project area you want change the editor representation. On the administration page of the selected project area click on Work Items.



On the Work Items management page select Types and Attributes.

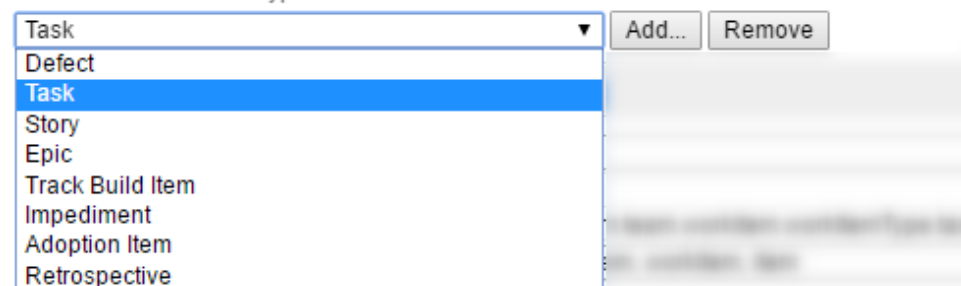


Choose the work item type for which you want to make the links visible. The following steps can be repeated for every work item type you like to visualize the Stages links. This example covers the work item type Task.

## Types and Attributes <sup>?</sup>

Define work item types, their attributes, and associate workflows and editor presentations

Choose the Work Item Type to edit:



Click on Add... under the attributes section.

Attributes <span style="float: right;">Add...</span>			
<input type="checkbox"/> Show only custom attributes			
Name	Type	ID	Actions
<input type="checkbox"/> Archived	Boolean	archived	N/A
<input type="checkbox"/> Comments	Comments	internalComments	N/A
<input type="checkbox"/> Corrected Estimate	Duration	correctedEstimate	N/A
<input type="checkbox"/> Created By	Contributor	creator	N/A

Add a custom attribute representing the Stages link.

**Add Attribute** ✕

Create Attribute

Name: \*

ID: \*

Type:  ▼

Reuse Existing Attribute

Attribute:  ▼

Property	Description
Name	<AttributeName> (e.g. Stages Link)
Id	de.methodpark.stages.rtc.attributes.processGuidanceId
Type	Medium HTML

The added attribute should now be visible in the attributes section of the work item type view.

Attributes <span style="float: right;">Add...</span>			
<input type="checkbox"/> Show only custom attributes			
Name	Type	ID	Actions
<input checked="" type="checkbox"/> Stages Link	Medium HTML	de.methodpark.stages.rtc.attributes.processGuidanceId	
<input type="checkbox"/> Archived	Boolean	archived	N/A

Now, as we have created the new attribute for the work item types, this attribute has to be added for the editor views. To do so, click on the Editor Presentations.

## Work Items

[Types and Attributes](#)

[Enumerations](#)

**[Editor Presentations](#)**

[Workflows](#)

[Change Management Type](#)

[Binding](#)

Choose the editor presentation for which you want to make the changes. The following steps can be repeated for every editor presentation you like to visualize the Stages links. This example covers the editor presentation `com.ibm.team.workitem.editor.task`.

### Editor Presentations ? Final (ignore customization of this data in child project areas)

Define the presentation structure of work item editors: pages, sections, and presentations. Drag to move sections or presentations.

Choose the Editor Presentation to edit:

com.ibm.team.workitem.editor.default ▼ Add... Remove

- com.ibm.team.workitem.editor.default
- com.ibm.team.workitem.editor.task**
- com.ibm.team.appt.editor.story
- com.ibm.team.workitem.editor.retrospective
- com.ibm.team.workitem.editor.impediment
- com.ibm.team.rtc.editor.adoption
- com.ibm.team.workitem.editor.buildtracking
- com.ibm.team.appt.editor.epic
- com.ibm.team.appt.planPreview
- com.ibm.team.workitem.web.inline
- com.ibm.team.workitem.lightweight

In the editor presentation look for the Details slot and click on the green plus-sign to add a new presentation. Select the attribute you created before, set the kind to Small HTML and confirm.

**Add Presentation**

Attribute-based Presentation

Attribute: \*

Kind: \*

Non-Attribute-based Presentation

Kind:

Label:

Description:

ID:

Properties:

Key	Value	Actions
-----	-------	---------

The new attribute field should now be visible in the editor presentation.

The screenshot shows a 'Details Slot' window with a 'Details' header. The form contains the following fields:

- Type: [Dropdown]
- Filed Against: [Dropdown]
- Project Area: ABC Project
- Team Area: XYZ Team
- Creation Date: [Date Picker]
- Created By: [Dropdown]
- Tags: [Text Area]
- Owned By: [Dropdown]
- Priority: [Dropdown]
- Planned For: [Dropdown]
- Estimate: [Text] Correction: [Text]
- Time Remaining: [Text]
- Due Date: [Date Picker]
- Resolution Date: [Date Picker]
- Resolved By: [Dropdown]
- Stages Link: [Text] (highlighted with an orange border)

## Operations and Workflows in RTC

### Operations

Operations are predefined and extensible events on the RTC Server. For each operation several precondition and follow-up actions can be assigned. A Precondition means that the execution of this action is synchronous and the triggering mechanism has to wait until the action is finished. In contrast a Follow-Up condition indicates that the action is executed in the background.

The Stages RTC Plug-in is an extension to such operations.

### Precondition & Follow-Up

As mentioned above, actions bound to certain operations can be separated in precondition and follow-up actions. What this means is that the workflows, which were exported to a project area on the RTC System can be executed in two ways:

#### *Precondition*

Workflows defined as preconditions are executed synchronously, before any changes are persisted on the RTC server. That means that this workflow can abort the incoming changes and inform the user about the situation.

#### *Follow-Up Action*

Workflows defined as follow-up actions are executed asynchronously as a background task. The workflow can alter the content of the changes and add or remove additional content to the work items or make configuration changes on the RTC server, but once the execution is finished the data is persisted. Due to the execution being asynchronous it is not possible to inform the user of its status.

If a script, which is associated to a workflow, should be executed on a specific operation and if it should be executed as a precondition or as a follow-up action, is determined by a simple comment at the beginning of the script.

The following table shows the currently implemented comments, which can be used in the Javascript files:

Operation	Precondition	Follow-up Action	Javascript Comment
Save Work Item	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<code>// STAGES_SAVE_ITEM_PRECONDITION</code>
Save Work Item	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<code>// STAGES_SAVE_ITEM_FOLLOWUP</code>
Save Change Set Links and Comments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<code>// MODIFY_CHANGE_SET_PRECOND</code>
Save Change Set Links and Comments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<code>// MODIFY_CHANGE_SET_FOLLOWUP</code>