

---

## Lab 3    Configuring Work Items

This lab will guide you through the definition of a new Work Item Type in Rational Team Concert. You will define the basic structure and information for this new type. The lab will focus on how these items can be described in your process using Rational Method Composer, and how to realize this information description in a Rational Team Concert process template.

### Lab Scenario

The team working in “Nifty Application 1.0” would like to schedule and track results of meetings using Rational Team Concert work items.

In this lab you will:

- Configure an existing process to describe how meetings will be managed.
- Create a new work item type from the process description
- Use the work item type in a work item template.
- Elaborate the work item type with new attribute types and reuse existing attributes
- Create and understand work item's workflows, their description in the process, and how to link workflows with work items.
- Adjust the presentation of your work items.

Lessons Learned:

- Describe workflows in your process
- Communicate process description changes to Rational Team Concert members who will adopt them
- Create work item types and customize all the basic elements involved in their definition

### 3.1 Update the process description

You are going to add content to your tailored process in RMC. Customizing the process description for your team organization process in Rational Method Composer is a task performed by the Process Engineer role.

#### Reload the RMC Workspace Library



If you shut down RMC, you will need to re-load the RMC workspace library when you restart.

Select **File>Open>Method Library**. In the Open Method Library window, select **Open method library from workspace**, and then select **Finish**.

The process customization tasks that you will perform are:

- Create a new Work Item Type that represents the workflow for meetings.

#### Process customization for the lab



The main purpose of this workshop is to guide you through process customization in Rational Team Concert and how this can be driven by the enactment of your enterprise process description performed in Rational Method Composer.

It is for this reason that we provide you with Rational Method Composer method plugins with the basic process content we will need for the different lab scenarios; to put you in the context for the RTC process customization exercise. In your real environment you, as process engineer, begin by creating this process description in RMC.


#### 3.1.1 Complete the process description of the new Work Item Type

The Nifty team will use the new work item type to manage meetings. You will complete the description of this work item type in your process.

- \_\_1. Document the work item type in RMC
  - \_\_a. Switch to RMC. In the Authoring perspective, in the Library view, navigate to *practice > mgmt > scrum > extend-acme > Method Content > Content Packages > acme*

- \_\_b. Right click and select **New > Work item type**

**User-defined types**



The “Meeting” work item type is not a standard type in RMC. This type has been added to this library using the user-defined types feature.

To see how this type was added, click *File > Create UDT Definition*. View the XML then close the window. For more information, refer to RMC online help and the RMC and Practices community on DeveloperWorks.

- \_\_c. Enter:


- \_\_i. Presentation name: Meeting
- \_\_ii. Name: meeting (if preferences set correctly, this will be automatically set.)

- \_\_d. Add a predefined state diagram to the main description

- \_\_i. Click the rich text editor icon to the left of Main Description.

**▼ Detail Information**

Provide detailed information about this work item type.



Main description:

- \_\_ii. Click the Add image link
- \_\_iii. Browse to add *resources > meeting\_states.jpg*  
(Full path: *root\PEW\Workspaces\Lab1\practice.mgmt.scrum.extend-acme\guidances\practices\resources\meeting\_states.jpg*)
- \_\_iv. Click **OK**
- \_\_v. Add additional text as desired.
- \_\_vi. Click **Save**.

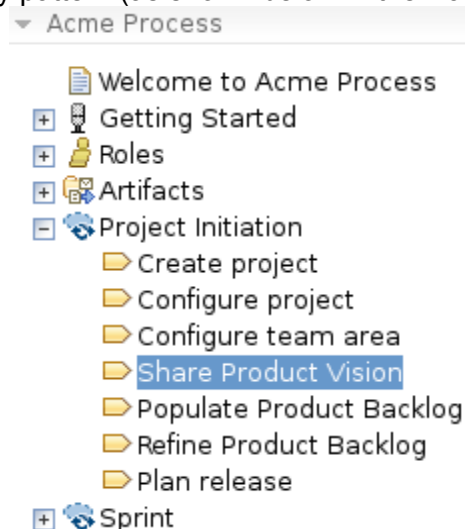
**What does this description mean?**

In this lab you will create a Work Item Type in Rational Team Concert representing this RMC element.

The text and diagram in the description of the RMC element gives information about the main actions taken by the person responsible for a work item of this type. This is the source information you use to define the RTC workflow of the work item type later.

- \_\_2. Open the *Pending Changes* view. Right-click on the *Process Extensions > Outgoing* folder and select **Check in and deliver**. (If you are prompted to create a work item, create one).
- \_\_3. Use the new work item type in a work item template
  - \_\_a. In the library view, open practice > mgmt > scrum > extend-acme > method content > content packages > acme > tasks > share\_product\_vision.acme (created in Lab 2)
  - \_\_b. Click on the *Relationships* tab (it may be elided, in which case you need to click the *Show List* icon to the right of the tabs).
  - \_\_c. Scroll down to the *Of Type* relationship section.
  - \_\_d. Click **Add**
  - \_\_e. Type *meeting* to filter the selections.
  - \_\_f. Click *meeting*. Click **OK**. Click **Save**.

This change affects every place the task is used, including the Project Initiation workflow/capability pattern (as shown below in the Acme process website).



### Capability patterns and work item templates



Whenever you want to describe a flow of tasks in RMC, you use work breakdown structures, called Capability Patterns (for partial processes) or Delivery Processes (for complete process workflows).

These work breakdown structures can be used to create work item templates in RTC. When you create work items from such a work item template, you get:

- a work item for each task
- a link from each work item to the task in the RMC website from which it was created.

### 3.1.2 Publish the new process description

Now you want to publish this new version of your process to make it available to all the teams working with Acme Process.

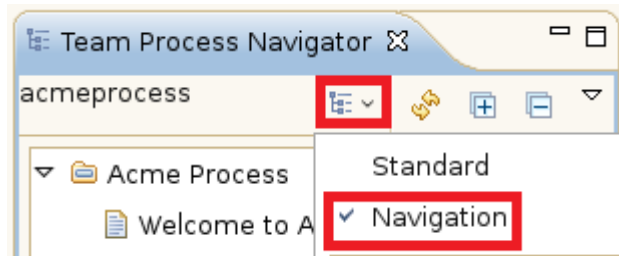
- \_\_1. Publish the changes to the server.
  - \_\_a. Under the Configuration menu, select **Publish...**
  - \_\_b. Make sure *Use publish options stored in the method configuration* is **checked** and Don't review publish options is selected (this will assure that the process will publish with the options defined in the configuration). Select **Next**.
  - \_\_c. Select **acmeprocess** in Method configuration and **Next**.
  - \_\_d. Make sure the option **Java EE web application** is selected with the following options, then select **Finish**. (Select **OK** when prompted to overwrite)
    - \_\_i. Directory: root/PEW/Downloads/acmeprocess
    - \_\_ii. Web application name: acmeprocess3
    - \_\_iii. Package format: WAR
    - \_\_iv. Leave other settings as defaulted.
  - \_\_e. Make sure the Publishing Report shows no errors then close the window.
  - \_\_f. Move the acmeprocess3.war file from root/PEW/Downloads/acmeprocess to root/PEW/IBM/JazzTeamServer/server/tomcat/webapps; (Replace any existing acmeprocess.war file).
- \_\_2. Verify the changes
  - \_\_a. In your browser navigate to <https://clm.process.ws/acmeprocess>.
  - \_\_b. In the treeview, select *Rational Team Concert > Work item types > Meeting* and review the information.

## 3.2 Customize the process in RTC

We will now update the Nifty Application Project area to match the changes in the process description.

### \_\_1. Create the Work Item type

- \_\_a. Go back to RMC and switch to the Team Process perspective
- \_\_b. Choose acmeprocess in the Method drop-down field
- \_\_c. On the *Team Process Navigator*, select the option to display the treeview in Navigation mode.



- \_\_d. In the treebrowser, right click *Acme Process > Rational Team Concert > Work Item Types > Meeting*
- \_\_e. Click **Create Jazz Work Item Type**
- \_\_f. Choose the following values and click **Finish**:
  - \_\_i. Project Area: Nifty Application Project (within the SoftwareDev repository)  
(Note: We are creating this in a “real” project to try it out. Later we will harvest this into ccm-pew and distribute it as a template to be used by new projects.)
  - \_\_ii. Context root: /acmeprocess3  
(Reminder: context root is the URL where process guidance is published).
  - \_\_iii. Workflow: Defect Workflow (leaving as default – we can update this later in RTC)

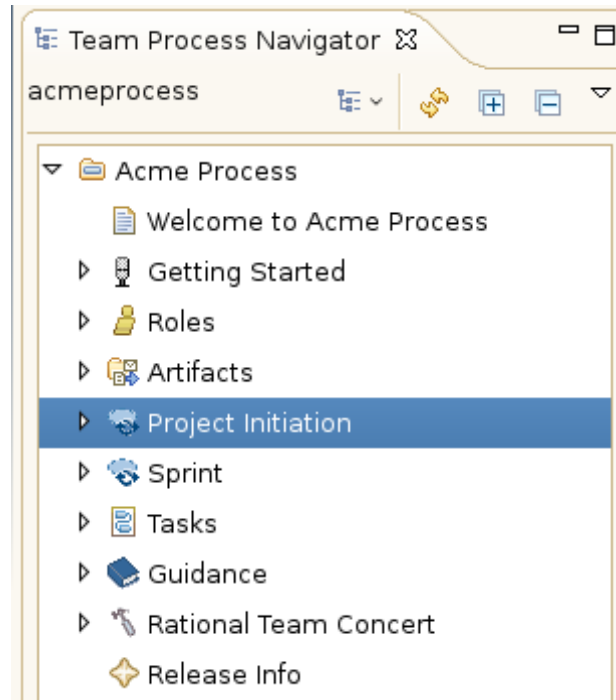
#### Administrative override?



Don't be concerned if you see messages that say “Permission denied” followed by “Administrative override”. Creating work item types and templates from RMC requires administrative privileges. RTC produces these messages as part of the interface.

### \_\_2. Replace the work item template for Project Initiation

- \_\_a. Still in the Team Process Perspective, right click on Acme Process > *Project initiation*.



\_\_b. Click **Create Jazz Work Item Template**.

\_\_c. Choose the following values and click **Finish**:

\_\_i. Project Area: Nifty Application Project (within the softwareDev repository)

\_\_ii. Context root: /acmeprocess3

\_\_3. Make further changes to the work item type using RTC

\_\_a. In the browser, you should be in the Nifty Application Project area. Click the wheel icon at the top right, and click Manage this Project Area.

\_\_b. Click the *Work items* tab on the left.

\_\_c. Under *Choose the Work Item Type to edit*, click *Meeting*.

### What is the deal with the Ids?

When customizing your process you will have to define IDs for different items: work item types, attribute types, presentations, ...



It is a good practice to define a naming convention within your enterprise. The elements of process customizations should follow this naming convention so the RTC process is consistent in your company, and it will ease the development leveraging these IDs: you will be able to quickly locate process customizations within the whole process. In addition, as removing elements is an operation to avoid for its potential impact, it is a good practice to modify the ID of the elements you want to deprecate with a certain pattern instead of just deleting the elements.

### What is a Work Item Type Category?



Work Item Type Categories define a common workflow and custom attributes shared by all the work item types that belong to it.

As our new work item type will have its own workflow, we need to create a new category along with the type.

\_\_\_d. Click the *Icon: Edit* button and select an icon, such as **adoption.gif**



- \_\_\_e. **Save** your changes.
- \_\_\_4. Now we will add some custom attributes.
- \_\_\_a. Navigate to the **Attributes** section of the page.
  - \_\_\_b. Select **Add ...**
  - \_\_\_c. Enter the following values:
    - \_\_\_i. Name: Meeting minutes
    - \_\_\_ii. ID: com.acme.workitem.attribute.minutes
    - \_\_\_iii. Type: Large HTML
    - \_\_\_iv. Select **OK**.
  - \_\_\_d. Add another attribute:
    - \_\_\_i. Name: Affected Departments
    - \_\_\_ii. ID: com.acme.workitem.attribute.affdepts
    - \_\_\_iii. Type: String List
    - \_\_\_iv. Select **OK**.
- \_\_\_5. Reuse existing attribute:
- \_\_\_a. Select **Add ...**
  - \_\_\_b. Check Reuse Existing Attribute and select “Affected teams” from the drop-down list
  - \_\_\_c. Select **OK**.

**Re-using attributes**

This reuse feature allows you to leverage attribute customizations from other work item type categories, easing maintenance of your process.

- \_\_\_6. Define enumeration for the formality attribute:
- \_\_\_a. Click the **Enumerations** tab
  - \_\_\_b. Select the **Add...** button located below *Choose the Enumeration to edit*
  - \_\_\_c. Enter the following values and select **OK**:

- \_\_\_i. Name: Formality
- \_\_\_ii. com.acme.workitem.enum.formality
- \_\_\_iii. Process Specification: CHECKED
- \_\_\_d. Select **Add...** at the *Literals* section, and enter the following values for the “Name” attribute of each literal value, you can leave the “External Value” field empty: (assign icons as you wish)
  - Low
  - High
- \_\_\_i. Select the value **Low** for *Default Literal* and *Unassigned Literal*
- \_\_\_ii. Click **Save**
- \_\_\_e. Create the attributes for the enumeration:
  - \_\_\_i. Click *Types and Attributes* tab
  - \_\_\_ii. Make sure *Meeting Work Item Type* is selected.
  - \_\_\_iii. Navigate to the *Attributes* section of the page.
  - \_\_\_iv. Select **Add ...**
  - \_\_\_v. Enter the following values, then select **OK**:
    - \_\_\_a. Name: Formality
    - \_\_\_b. com.acme.workitem.attribute.formality
    - \_\_\_c. Type: Formality  
Do not select *Formality (enumeration list)*.  
*Formality* let's you assign just one formality value for a meeting.  
*Formality (enumeration list)* lets you assign multiple formality values.

#### Adding attributes to existing work items



If you add new attributes to work item types that already have work items created, you have to Synchronize the attributes to be able to use the new attribute in that existing work items.

The article *Cool “Hidden” Features in Rational Team Concert: Part 2* explains this feature. [Part 3](#) contains links to the other articles (attributes synchronization is located in [part 2](#) of the series)

- \_\_7. Define the workflow: the Process Engineer described the workflow in RMC. Now you will create a work item workflow matching that description.
- \_\_a. Select the *Workflows* tab.
  - \_\_b. Select **Add...** next to the *Choose the workflow to edit* field.
  - \_\_c. Enter the following values and then select **OK**:
    - \_\_i. Name: Meeting Workflow
    - \_\_ii. ID: meetingWorkflow
  - \_\_d. Navigate down to the States section. Click **Add...** button and create the states with the following information: (assign icons as you will).

Name	Group	Show Resolution
Planned	Open	Unchecked
Scheduled	Open	Unchecked
Minuted	Open	Unchecked
Actions addressed	Open	Unchecked
Closed	Closed	Unchecked
Canceled	Closed	Checked

### What are the State Groups?

The State Groups help you categorize states and resolutions that relate to each other.



You can create new state groups and associate them to the default group categories: Open, In-Progress, Closed. These grouping of states and resolutions are very useful for building queries / reports, and when you extend Rational Team Concert and want to interact with Work Items states.

In addition, these categories are associated with OSLC state groups, thus providing a way of isolating the detailed knowledge of your actual defined workflow for integration with other systems via OSLC.

- \_\_e. Create the actions that cause the state transitions: in the Actions section of the page, perform the following:
  - \_\_i. Click **Add**
  - \_\_ii. Enter *Plan* as the name. For *Target state* click **Planned**. Select OK.

\_\_iii. Repeat steps i – ii, with the rest of these actions.

<b>Action name</b>	<b>Target state</b>
Schedule	Scheduled
Complete minutes	Minuted
Close actions	Actions addressed
Close	Closed
Cancel	Canceled

\_\_f. Describe the legal transitions (from state to state): navigate to the Transitions section of the page and perform the following changes:

\_\_i. In the combination From “Planned” To “Scheduled”, click the dropdown and select *Schedule*.

\_\_ii. Repeat with the rest of the transitions until you build the following transition table (which maps to the flow you described in Rational Method Composer earlier in this lab):

Transitions							
From	To	<input type="checkbox"/> Planned	<input type="checkbox"/> Scheduled	<input type="checkbox"/> Minuted	<input type="checkbox"/> Actions addressed	<input type="checkbox"/> Closed	<input type="checkbox"/> Cancelled
<input type="checkbox"/> Planned		<None>	Schedule ↕	None ↕	None ↕	None ↕	Cancel ↕
<input type="checkbox"/> Scheduled		Plan ↕	<None>	Complete minutes ↕	None ↕	None ↕	Cancel ↕
<input type="checkbox"/> Minuted		Plan ↕	None ↕	<None>	Close actions ↕	None ↕	Cancel ↕
<input type="checkbox"/> Actions addressed		Plan ↕	None ↕	None ↕	<None>	Close ↕	Cancel ↕
<input type="checkbox"/> Closed		Plan ↕	None ↕	None ↕	None ↕	<None>	None ↕
<input type="checkbox"/> Cancelled		Plan ↕	None ↕	None ↕	None ↕	None ↕	<None>

\_\_g. Define the start action:

\_\_i. Select the dropdown next to Start Action and click **Plan**.

### Resolve and Reopen actions



You may wonder why we left these fields blank. Our defined workflow has two different possible end states depending on the path: we didn't consider a unique and global resolve action.

In addition, we defined a reopen type of action (“Plan”) for each of the states. We could go back into RMC and correct the diagram, but often it is best to just show the normal flow, and not the less usual transitions, such as reopening/replanning.

- \_\_h. Define the resolutions: you want to further explain the possible reasons for cancelling a meeting.
  - \_\_i. Navigate to the Resolutions section.
  - \_\_ii. Click **Add...**
  - \_\_iii. Enter *No longer required* as the name. Select **OK**.
  - \_\_iv. Repeat the steps to include a resolution called *Duplicate*.
- \_\_i. Navigate back to the *Actions* section of the page and configure resolutions:
  - \_\_i. In the row for the *Cancel* action, click on *Click to select Resolutions*.
  - \_\_ii. Click **Add All**. Click **OK**.
- \_\_j. **Save** your changes.
- \_\_k. Associate the workflow to the work item:
  - \_\_i. Navigate to the **Types and Attributes** tab.
  - \_\_ii. Choose *Meeting* under *Choose the Work Item Type to edit*.
  - \_\_iii. In the *Workflow* dropdown, select **Meeting Workflow**.
  - \_\_iv. **Save** your changes.
- \_\_8. Customize the presentation for the work item

**Work Item editor presentation structure:**

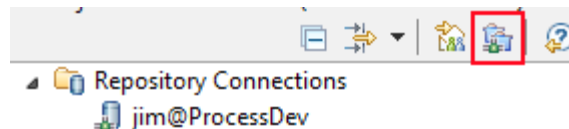
The work item editor presentation is structured in **tabs**. Each tab contains **slots** which are the containers for the actual presentation of information. This information can be “attribute-based” or “non-attribute” based (like the presentation for the attachments of a work item).



How information is placed in a tab, and the organization of the slots, is based on the concept of a **layout**. There are different types of layouts depending the information you want to place and how you want to organize it. For more information see [Work item editor presentations](#).

The presentation editor is based on the concept of reuse. The definition of tabs and slots can be shared among different work item editor presentations. If you modify a shared configuration it will apply to all the presentations that make use of it. As an alternative, you have the option of creating new presentations (whether tabs or layouts), or duplicating an existing presentation and customize for your particular needs.

- \_\_a. For ease of use, you will create a presentation based on an existing one, customizing it to hold the required attributes.
  - \_\_i. Back in RMC, change the perspective to *Work Items*.  
(This is a Jazz/RTC perspective that happens to also be available in RMC).
  - \_\_ii. Click on *Manage Connected Project Areas*



- \_\_iii. Select the existing repository connection then click **Next**
- \_\_iv. Select all project areas then click **Finish**.
- \_\_v. Change the view to *Team Artifacts*.
- \_\_vi. Right-click *Nifty Application Project [SoftwareDev]* and click **Open**.
- \_\_vii. Click the *Process Configuration Tab*.
- \_\_viii. Click *Project Configuration > Configuration Data > Work items > Editor presentations*.

- \_\_ix. Make sure in the *Choose editor presentation to edit* box the `com.ibm.team.workitem.editor.default` is selected, and select **Duplicate ...**
- \_\_x. Enter the following ID and select **OK**: `com.acme.workitem.editor.meeting`
- \_\_xi. Click **Save**.

#### Eclipse client vs. browser client



Both RMC and RTC Eclipse clients allows for Jazz process template customization. Although most features are available in both Eclipse and browser clients, some features for modifying presentations is only available in the Eclipse client, so we have chosen to use this interface for this lab.

- \_\_b. Adjust presentation of attributes: you based your presentation in the default one which is used by tasks. As the Meeting work item is a specialized type of task, you will be just adding a specialized tab for the new information and reusing the rest of the information.
  - \_\_i. Click **Add Tab ...** and enter the following values:
    - \_\_a. Title: Meeting Information
    - \_\_b. Layout: Custom Attributes Layout
    - \_\_c. Create Tab ID: CHECKED
    - \_\_d. ID: `com.acme.workitem.tab.meetinginformation`
    - \_\_e. Click **OK**.
  - \_\_ii. High-light the new tab and select **Add Section ....** Enter the following values, then select **OK**.
    - \_\_a. Title: Meeting summary info
    - \_\_b. Slot: Right
    - \_\_c. Create Section ID: CHECKED
    - \_\_d. ID: `com.acme.workitem.section.meetingsummary`
  - \_\_iii. Repeat the previous steps to add a new section with the following details:
    - \_\_a. Title: Meeting minutes
    - \_\_b. Slot: Left
    - \_\_c. Create Section ID: CHECKED

\_\_d. ID: com.acme.workitem.section.meetingminutes

- \_\_iv. Highlight *Meeting Summary Info* section and click **Add Presentation...** for each of the items listed below. (do it in the order of appearance. All of them are “Attribute based”):

Attribute	Kind	Label	Description
Formality	Enumeration	NA	NA
Process Guidance (rmc_guidance_attribute._rmc.meeting)	HTML	NA	Link to workflow description in process website.
Affected Departments	String List	NA	NA
Affected Teams	Process Area List	NA	NA

- \_\_v. Now highlight *Meeting Minutes* section and click **Add Presentation...** for each of the items listed below (do it in the order of appearance):

Attribute	Kind	Label	Description
Meeting minutes	HTML	NA	NA

#### Work item type process guidance



When RMC creates a work item type, it creates a *Process guidance* attribute that links back to the process guidance. If you look at other work item types in this template, you will notice that we used a different approach (using a non-attribute based presentation element). This other approach is described in the RMC community topic [Documenting work item types using RMC](#).

- \_\_c. Highlight the *Meeting Information* tab and click **Move Up** until you place it just after the *Overview* tab.
- \_\_d. **Save** your changes.
- \_\_e. Link the presentation to the work item:
- \_\_i. Navigate to the **Types and Attributes** node under *Work Items*.
  - \_\_ii. Choose *Meeting* under Work Item Types
  - \_\_iii. In the Work Item Editor box, select com.acme.workitem.editor.meeting
  - \_\_iv. **Save** your changes.



### Different types of editors?

You may noticed that there are other type of editors in this page. The main differences are:

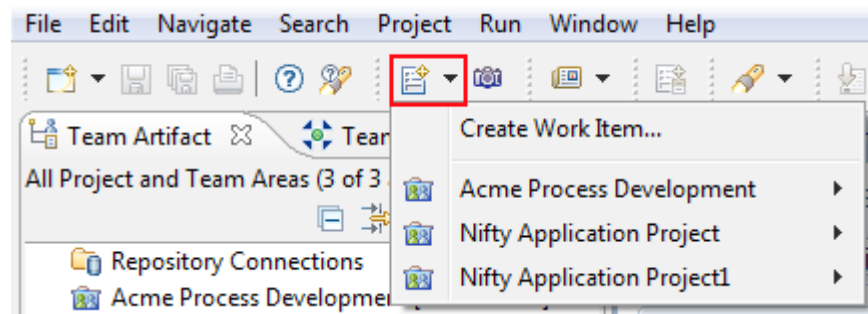


- Work Item Editor: the main editor for managing work items. Used in the different clients when you open a work item.
- Inline Work Item Editor: Inline editors are presented in the query results in the Web UI
- Lightweight Work Item Creation Dialog: presentation for wizards that allow for quick creation of work items, like when delivering change sets or from other integrated CLM applications.
- Plan Editor Preview: presentation for modifying work items from within a plan.

For all but the main editor, the default one usually is enough for the needs of your project, as they present the minimum basic set of attributes. However, you can also customize the other editors to your needs as we did in the example for the full editor presentation.

\_\_f. Check the presentation.

\_\_i. Select the arrow next to the New Work Item icon.



\_\_ii. Select **Nifty Application Project > Meeting**. A new work item appears

\_\_iii. Select the **Meeting Information** tab. It should look like the following:

\_\_9. Adjust permissions for attributes and workflow states: you have to give permissions to the roles for the new items created in the process.

\_\_a. Return to the project area editor and select **Permissions** under *Team Configuration* node.



Select under *Team Configuration*, not *Project*).

\_\_b. Check the option to **Show all actions and roles**

\_\_c. We will adjust permissions for meetings:

\_\_i. Adjust permissions to the attributes:

\_\_a. Open the node *Work Items > Save Work Item (server) > Modify the work item*

\_\_b. Click *Modify the 'Meeting minutes' attribute*.

\_\_c. Permit only the Scrum Master role to perform this action (remove the permissions from other roles).

\_\_ii. Adjust permissions for workflow:

\_\_a. Open the node *Work Items > Save Work Item (server) > Create a work item > Create a work item of a specific type*

- \_\_\_b. Under Create a 'Meeting' work item, **CHECK** the box for the Scrum Master and Product Owner roles, and **UNCHECK** for the rest of the roles.
- \_\_\_c. Open the node *Save Work Item (server) > Trigger a workflow action*
- \_\_\_d. Make sure the Scrum Master role is the only one with permission to perform the *Meeting workflow* actions to complete minutes, close actions, and close the meeting. All these workflow actions are part of the Meeting Workflow.

☐ Show actions by role ☒ Show all actions and roles

Actions	Everyone	Develop	Product Owner	Scrum Master	Stakeholder	Product
Stop working task workflow	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Schedule (Meeting Workflow)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Complete minutes (Meeting Workflow)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Close actions (Meeting Workflow)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan (Meeting Workflow)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Close (Meeting Workflow)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cancel (Meeting Workflow)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

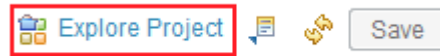
- \_\_\_d. **Save** your changes.

### 3.3 Try out the changes

We will now try out the new meeting work item type.

\_\_1. Create work items for Work Item type

\_\_a. In the web UI, click **Explore Project** to return to the Nifty Application Project project dashboard.



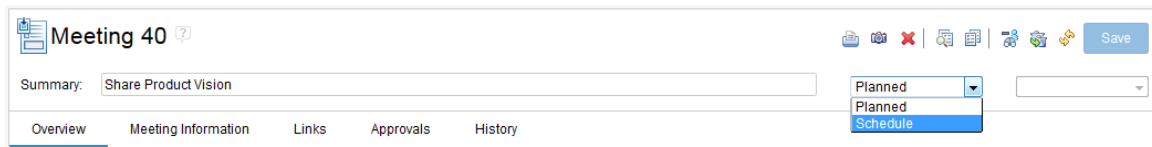
\_\_b. Click **Work items > Create from template**.

\_\_c. Click **Project Initiation** and then click **Create from template**.

\_\_d. Click the work item **Share product vision**.  
Observe that it is of type *Meeting*.  
Observe the new attributes that were added  
Observe (and optionally click on) the RMC process guidance link *See detailed description*.

\_\_e. Transition the work item through it's various states.

\_\_i. Click the *Planned* dropdown then click **Schedule**. Click **Save**.



\_\_ii. Continue transitioning states as you wish.

**More to do**

Normally we would also update the project dashboard to point to the new published process. Since this was done in Lab 2, we do not repeat that here.

### 3.4 Gather the process changes

After successfully customizing the process in a living project, you want to gather changes back into the templates managed in your process development environment and managed by Rational Method Composer, so new projects can be created with these new items already defined.

### Customizing and testing in my production project?

The configuration steps in this lab were directly performed in the Nifty Application production project to focus in the features introduced. In your real production environment you would have a testing environment in which you validate process changes before applying them in production. This test environment, depending on your needs, could range from just a test project area in the production server, or a complete test installation. In lab 2 the Process Engineer customized the process using a test project area before gathering the changes back into the organization's process template.

Replicating process customization changes from your test environment in you real production project area can be performed in any of these ways:



- The Project Lead replicates all the modifications from the test project area. For this operation, the enterprise wide naming convention for the IDs of process elements outlined as best practice previously in this lab will be very useful.
- If test and production project areas are always aligned, Project Lead can copy the entire process source XML from test and replace the one in production project area. This operation is to be performed with extreme care.

We have omitted this test project area intermediate step for the sake of workshop timing and to concentrate on the main operations of the lifecycle.

You can also deploy process definitions using Process Sharing, where a Provider Project allows Consumer Projects to use its process definition. We do not cover this feature in this workshop.

- \_\_1. Generate the process template from the Project Area:
  - \_\_a. Back in RMC, switch to the *Jazz Admin* perspective, and open the Team Artifacts view
  - \_\_b. Right-click the project area Nifty Application Project and select **Extract Process Template...**
  - \_\_c. Give it the following values and select **Finish**.
    - \_\_i. Name: Acme Nifty Based
    - \_\_ii. ID: acme.nifty.processtemplate.v2
    - \_\_iii. Summary: Nifty version 2.

**What if you want to reuse the template ID?**

You cannot create a process template from a Project Area and assign it an ID of a process that already exist in the repository. On the other hand, you can't delete a process template from the repository if a project area exist that was created based on it.



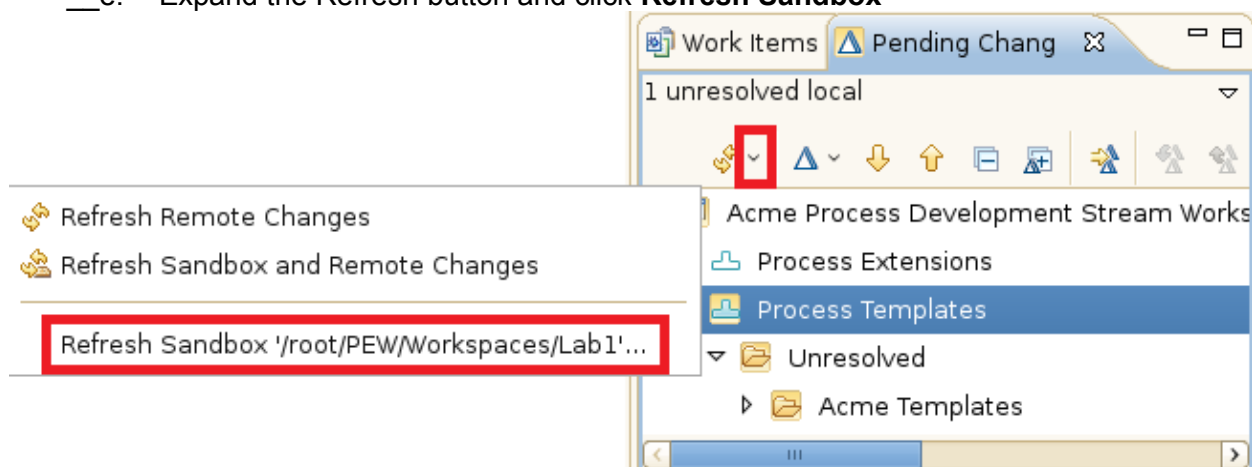
If you want to keep one corporate process template (and thus just one ID), you can generate a process template from a project area with the process shown here, export it and then re-import it giving it the existing ID; you will then be prompted to overwrite.

Alternatively, you can use the Process Sharing feature.

**Important!** A process template generated from a project area doesn't consider localization, but is generated in the language of the project area instantiation.

- \_\_2. Export the Process Template:
  - \_\_a. Back in the Process Templates view, right-click your newly created template called *Acme Nifty Based* and select **Export**.
  - \_\_b. Select *root/PEW/Downloads* as the directory and click **Finish**.
  - \_\_c. The file will be saved as *root/PEW/Downloads/template*
- \_\_3. Delete the temporary Process Template:
  - \_\_a. Right-click the *Acme Nifty Based* process template and select **Delete...**
  - \_\_b. Click **OK** when prompted
- \_\_4. Import the process template into the process development environment:
  - \_\_a. In the Process Template view, right-click the "jim@ProcessDev" connection and select **Import**.
  - \_\_b. Enter the following values in the Import wizard. Note that we are modifying the populated ID and Name:
    - \_\_i. Template Directory: CHECKED
    - \_\_ii. Template Directory: *root/PEW/Downloads/template*
    - \_\_iii. Name: *Acme Process V2*
    - \_\_iv. Process ID: *acme.process.v2*

- \_\_\_c. Select **Finish**. You have a new process template that contains the customizations performed via the Nifty project.
- \_\_\_5. Update the process template under SCM:
  - \_\_\_a. Right-click Acme Process V2 under the “ProcessDev” repository connection and select **Export**.
  - \_\_\_b. In Directory field enter: root/PEW/Workspaces\Lab1\Acme Templates(It is your SCM workspace!). Select **Finish**.
  - \_\_\_c. Change the perspective to *Work Items*.
  - \_\_\_d. Open the *Pending Changes* view as follows:
    - \_\_\_i. Click *Window > Show View > Other...*
    - \_\_\_ii. Click *Jazz Source Control > Pending Changes*
    - \_\_\_iii. Click **OK**
  - \_\_\_e. Expand the Refresh button and click **Refresh Sandbox**



Select **OK** when prompted so you refresh the whole sandbox (you can check just the specific folder for the process template if you want).

- \_\_\_f. Right-click the *Process Templates > Unresolved* folder in the Pending Changes View and click **Check-in and Deliver**. Type a comment (such as “Created V2”) and click **Finish**.

You have successfully updated your process development environment and the process templates with the latest changes.

**Final note: update template but keep localization**

In these last steps of the lab you saw how to update your corporate process template gathering changes from a project area. However, there's no easy way in Rational Team Concert to perform this process generation from a living project keeping the externalized strings localization information. See [Translatable Process Templates in Rational Team Concert 2.0](#)



If you want to gather changes from a Project Area keeping this localization information, you will have to manipulate the XML of the Process Template to manually replicate the changes. Some possible approaches:

1. Manually replicate the changes directly in the process template if you have clear where the XML has been modified
2. Generate a temporary project area based on your template and compare with the new XML. Note that some structural changes (like change in the order of XML tags), may be possible.
3. Update the process template as you did in this lab, and then re-do the steps to add localization to the process template.