Contact:

Method Park
Wetterkreuz 19a
91058 Erlangen
Germany

Phone
+49-9131-97206-550

Fax
+49-9131-97206-200

E-mail
stages-support@methodpark.de

Internet
www.methodpark.de

Version: 1.0

Date: 1.12.2017
## Contents

1 Terminology..................................................................................................................4  
2 Literature.....................................................................................................................5  
3 Stages Integration Guide............................................................................................7  
   3.1 Integration of an ALM System..................................................................................7  
   3.1.1 ALM System Configuration................................................................................7  
   3.1.2 ALM Server Configuration................................................................................7  
   3.1.3 ALM Server Properties.....................................................................................8  
   3.2 Exporting Stages Processes to ALM Systems.........................................................8  
   3.3 Install & Upgrade the Stages Processor..................................................................10  
4 Jira Integration Guide.................................................................................................11  
   4.1 Stages Plug-in Installation......................................................................................11  
   4.1.1 Jira Server Configuration..................................................................................11  
   4.1.2 Marketplace Installation..................................................................................11  
5 Stages ALM Modeling Guide......................................................................................12  
   5.1 Abstract Process....................................................................................................12  
   5.2 Concrete Process..................................................................................................12  
   5.3 Workflows............................................................................................................13  
   5.3.1 Elements of a workflow...................................................................................14  
   5.3.2 Relations between Workflow Elements...........................................................14  
   5.3.3 Modeling Workflows.......................................................................................15  
   5.4 Work Product Attributes......................................................................................16
1 Terminology

Instructions, warnings and other important notes are marked in this document with the following symbols:

- **Notice**: Shows an additional information
- **Warning**: Indicates that a problem can happen
- **Important**: Indicates an important aspect.
- **Danger**: Indicates that in special cases data loss may happen.

### Instructions

Used to show steps to perform

### Example

Shows an example

Related links are shown with this icon:

- **Related Links**
  
  *Terminology* on page 4
2 Literature

The following documents provide you with information about working with Stages:

### User Documentation

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editor Guide</td>
<td>The Editor Guide describes the handling of the Stages Web Editor.</td>
</tr>
<tr>
<td>Stages User Guide</td>
<td>The User Guide describes typical work processes which quality managers, project managers and developers carry out on a day-to-day basis using Stages.</td>
</tr>
<tr>
<td>Metric Guide</td>
<td>The Metric Guide describes how Stages metric reports can be created.</td>
</tr>
</tbody>
</table>

### Configuration/Administration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade Guide</td>
<td>The Upgrade Guide describes the required steps for upgrading to a newer version of Stages.</td>
</tr>
<tr>
<td>System Requirements</td>
<td>Describes the system requirements of the Stages system.</td>
</tr>
<tr>
<td>Metamodeling Guide</td>
<td>The Metamodeling Guide will provide you with assistance in defining an appropriate metamodel.</td>
</tr>
<tr>
<td>Administration Guide</td>
<td>The Administration Guide gives you guidance in the administration of Stages. Starting with the installation and customization to your company’s requirements, chapter two helps you to prepare Stages for the usage.</td>
</tr>
<tr>
<td>Issue Tracking Guide</td>
<td>The Issue Tracking Guide provides information about the definition of your specific issue tracking configuration.</td>
</tr>
<tr>
<td>PDF Printing Guide</td>
<td>The PDF Printing Guide describes how PDF documents can be created from different process elements.</td>
</tr>
<tr>
<td>People Application Guide</td>
<td>The People Application Guide provides information about activating your People Application and how to handle it.</td>
</tr>
<tr>
<td>Permissions Guide</td>
<td>The Permissions Guide describes in detail all the different permission settings.</td>
</tr>
<tr>
<td>Release Notes</td>
<td>The Release Notes describes the major enhancements and changes of the Stages version.</td>
</tr>
</tbody>
</table>
### Last Changes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Changes</td>
<td>The Last Changes document is a change log with detailed information about all changes made in this version.</td>
</tr>
</tbody>
</table>

### System Integration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIS Integration Guide</td>
<td>The ARIS Integration Guide describes how ARIS process models can be imported into Stages.</td>
</tr>
<tr>
<td>CMS Prefetch Guide</td>
<td>The CMS Prefetch Guide describes the caching configuration of the CMS status informations.</td>
</tr>
<tr>
<td>ClearCase Integration Guide</td>
<td>The ClearCase Integration Guide describes the integration with IBM Rational ClearCase.</td>
</tr>
<tr>
<td>PTC Integrity Integration Guide</td>
<td>The PTC Integrity Integration Guide describes the integration with PTC Integrity.</td>
</tr>
<tr>
<td>Subversion Integration Guide</td>
<td>The Subversion Integration Guide describes the integration with Subversion.</td>
</tr>
<tr>
<td>Sharepoint Integration Guide</td>
<td>The Sharepoint Integration Guide describes the integration with Microsoft Sharepoint.</td>
</tr>
<tr>
<td>LDAP Integration Guide</td>
<td>The LDAP Integration Guide describes how LDAP informations can be imported into Stages.</td>
</tr>
</tbody>
</table>

### Application Lifecycle Management Tools

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIRA Configuration Guide</td>
<td>This Guide describes how to configure the ALM Tool JIRA to work with imported Stages Processes.</td>
</tr>
<tr>
<td>RTC Configuration Guide</td>
<td>This Guide describes how to configure the ALM Tool RTC to work with imported Stages Processes.</td>
</tr>
</tbody>
</table>

These documents are delivered with Stages. They can be accessed via the Stages Info page.
3 Stages Integration Guide

The following document details the integration of Stages into an Application Lifecycle Management (ALM) service. This process includes altering the configuration of Stages to enable Processes modeled in Stages to be exported to the chosen ALM system.

3.1 Integration of an ALM System

The configuration of a ALM system needs to be done in the enactment section of PKitConfig.xml (see example configuration below).

```xml
<enactment>
  <enactment-system name="rtc">
    <enactment-server ident="rtc1" url="http://rtc.example.net/ccm"
      displayName="RTC Server">
      <enactment-property name="username" value="admin" />
      <enactment-property name="password" value="secret" />
      <enactment-projects>
        <enactment-project name="TEST1" />
        <enactment-project name="TEST2" />
      </enactment-projects>
    </enactment-server>
  </enactment-system>
  <enactment-system name="jira">
    <enactment-server ident="jira1" url="https://jira.example.net/"
      displayName="JIRA Server">
      <enactment-property name="username" value="admin" />
      <enactment-property name="password" value="secret" />
      <enactment-projects>
        <enactment-project name="TEST1" />
        <enactment-project name="TEST2" />
      </enactment-projects>
    </enactment-server>
  </enactment-system>
</enactment>
```

3.1.1 ALM System Configuration

For each ALM system type that shall be accessed by Stages an enactment-system section must be defined under the appropriate enactment section in PkitConfig.xml. You have to provide a value for the attribute name.

- As name you have to specify either jira or rtc depending on the type of system you are using.

3.1.2 ALM Server Configuration

For each ALM server that shall be accessed by Stages an enactment-server section must be defined under the according enactment-system section in PkitConfig.xml. You have to provide values for the attributes ident, url and displayName.

- As ident you have to specify either the same value as for the host name or another unique identifier string. ALM servers will later reference this ident.
- The url must point to the exact location where the ALM system is running. The URL can contain the protocol (http or https), the server's host name, and optionally a port number and the path to the ALM.

Examples:
- http://alm.example.net/company/alm
- https://alm.example.net:8080
- The displayName is the name that is displayed in Stages CMS during export.
3.1.3 ALM Server Properties

Some properties can be defined for a specific ALM server (`enactment-server`), in `enactment-property` elements.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td></td>
<td>You may enter the ALM username that will be used by Stages if during export no explicit user and password is given.</td>
</tr>
<tr>
<td>password</td>
<td></td>
<td>You may enter the password for the ALM user that is used by Stages if during export no explicit user and password is given.</td>
</tr>
<tr>
<td>trust-all-ssl</td>
<td>false</td>
<td>When set to <code>true</code>, disables any verification checks for certificates in case of SSL-secured HTTPS connections. It is recommended to only use this in test environments where no valid certificates exist (e.g. self-signed certificates).</td>
</tr>
</tbody>
</table>

3.2 Exporting Stages Processes to ALM Systems

This section shows, step by step, how to export a stages process to a configured enactment system.

**Warning:** To perform the following actions the Stages installation must have the feature for the corresponding enactment system enabled and the user has to have the permissions to do Enactment Exports. The necessary permissions can be found in the Permissions Guide.

### Instructions

The following steps are necessary for exporting a Stages process to an enactment system:

1. First of all navigate to the start page of the process you would like to export.

2. Select the menu command "Actions→Import/Export"

3. Click on the icon for the Enactment export:
4. On the enactment export dialog you can specify the export target and have the following choices:

- Select the enactment system from the dropdown menu ①.
- In the dropdown menu ② you can select a server of the enactment system type selected above. If you want to enter a custom server address you must select the ”Custom” entry from the dropdown menu. After that, you can enter the server url in ③.
- The project field ④ defines the target project that exists in the enactment system. Similar to the server URL you can specify a custom project. Therefore select the ”Custom” entry in the dropdown menu and enter your custom project name in ⑤.
- The radio buttons ⑥ let you choose between predefined credentials and custom credentials for authentication on the target enactment system. If you check the ”use custom credentials” button, you can enter your username and password for the export in ⑦.
- It is also possible to export legacy Stages processes to Rational Team Concert by activating the checkbox at ⑧.
- To perform the export to the specified enactment system click on ”Export” ⑨.

**Note:** For instructions on how you can define enactment export targets, so that they are available in the dropdown menus, please see *Integration of an ALM System* on page 7.

5. After executing the export you get a result dialog similar to the one in the figure below.

With this dialog you can verify the following:

- The kind of enactment system ①.
- The server that received the export ②.
• The server URL ③.
• The username that was used to authenticate at the target server ④.
• The project area into which the stages workflow was exported ⑤.
• A result message which gives information about the executed export ⑥. This message gives human readable information about any errors which may have occured.

### 3.3 Install & Upgrade the Stages Processor

This section shows step by step how to upgrade the Stages-Processor in an existing Stages installation.

⚠️ **Warning:** Only upgrade Stages-Processor if instructed by the Stages-Support, not every Stages-Processor version is compatible with every Stages version.

#### Instructions

The following steps are necessary to upgrade the Stages-Processor:

1. Stop Stages
2. Replace the existing stages-processor.war with the new version.
   
   This file can be found under: `<Stages-Install-Directory>\tomcat\webapps`

   ⚠️ **Warning:** Please verify, that no copy or backup of the original file remains in the folder.

3. Remove any stages-processor folder in `<Stages-Install-Directory>\tomcat\webapps` if it exists. Only the corresponding .war file has to be in that folder.

4. Start Stages
This chapter tells you how to configure Jira for automations coming from exported Stages processes. To get things work, you have to make adjustments to the setenv.sh/setenv.bat file on the server and install the Method Park Stages™ Plugin for Jira.

4.1 Stages Plug-in Installation

4.1.1 Jira Server Configuration

The following section explains, how to prepare the Jira Server for the Plugin installation.

Instructions

To prepare the Jira Server, follow these steps:

1. Go to the bin-folder of your Jira installation (<jira-install>/bin).
2. Open setenv.sh on Linux or setenv.bat on Windows for editing.
3. Find the section JVM_SUPPORT_RECOMMENDED_ARGS and add the following parameters.
   There must be no space between the comma separated values of bootdelegation:
   - Datlassian.org.osgi.framework.bootdelegation=sun.*,com.sun.*,jdk.internal.*,
   - jdk.nashorn.*
4. Restart your Jira server.

4.1.2 Marketplace Installation

The Method Park Stages™ Plugin for Jira is delivered via Atlassian Marketplace.

Instructions

To install the plug-in, follow these steps. If you don't have sufficient permissions, please contact your Jira administrator.

1. Login to your Jira server.
2. Go to Jira administration page and select the "Add-ons" page.
3. Search the marketplace for "Method Park Stages Plugin".
4. Press the installation button next to the Stages Plugin.
5. After the successful installation refresh the page. You will find the new subsection "STAGES PLUGIN".
6. Go to the "Status" page of the "STAGES PLUGIN" section.
7. Make sure all statuses are OK.
5 Stages ALM Modeling Guide

This guide describes, how Stages processes, provided for exports to enactment systems like Jira and Rational Team Concert, should be modeled. The Stages process can tell an enactment system, how to react on certain events and defines a specific behavior for this event.

A part of a Stages process, which is responsible for the ALM System to react on such events, is called workflow and usually consists of an activity, a work product and a script.

This guide provides guidelines for extending your existing Stages processes with so-called workflows to bring the process to life on the enactment system. The parts of a workflow are explained in the chapter Workflows on page 13. After describing the different components of a workflow, the chapter Abstract Process on page 12 gives an idea of how Stages processes should be structured to avoid losing clarity. The chapter Concrete Process on page 12 describes a process which was extended by workflows in detail.

5.1 Abstract Process

To make your process model easier to read and understand, we suggest creating an abstract process model at first. This model only consists of activity groups. Every activity group has one responsible role. The following figure shows an example of a process made of activity groups:

As you can see in this image, this example has three activity groups (Define, Build and Test). Each group represents a different part of the process and can contain a various number of workflows.

In this perspective, we do not see details about the automation on the enactment system but the reader of the process gets a first idea about the structure of the process. After creating this simple process, we will have a closer look at the activity group itself in the next chapter.

5.2 Concrete Process

Figure 2 on page 12 shows a simple process defined in the activity group Build of our abstract process.

We have modeled a simple process for developing software. Following this process, we create source code from incoming work packages. Writing code generates software. To verify the result, the process tells us to create reviews. That means for instance, that a developer has to create review items on the ALM System, every time he finishes his work on a work package.

With the ALM Integration Plug-in for RTC or Jira, Stages can instruct the Application Lifecycle Management Tool to create these reviews automatically, when a specific event occurs.
Using this example as starting point, we extend the model with some additional elements.

As you can see in the figure above, we added a transition of the work package from a state "In Progress" to "Done" when the activity "Write Source Code" is finished. In addition, this automation needs some instructions in the form of a script. These scripts are executed on the enactment system and consist of javascript code built against a specific ALM Plug-in API. Specific means that the different supported Systems (e.g. RTC, Jira) have their own javascript API. To inform the ALM System about these instructions in a script, we have to assign this script to the corresponding association. In this case, the output association, because we want the system to execute the script when a work package transitions its state from "In Progress" to "Done".

The content of the script and the scripting API in general is not part of this chapter. But for further information on this topic see the Scripting Guide. The resulting model after associating the script to the output work product is illustrated in Figure 4 on page 13.

This example is also called a "Workflow" and is explained in the chapter Workflows on page 13 in more detail.

5.3 Workflows

As described in the chapter above (Concrete Process on page 12), workflows are used to define actions on the ALM System. Below you can find detailed information about workflows. In the first part, we have a closer look at the Stages elements a workflow is made of (Elements of a workflow on page 14). The chapter Relations between Workflow Elements on page 14 describes how these elements have to be connected between each other in order to be considered as workflow for an ALM System.
5.3.1 Elements of a workflow

A simple workflow consists at least of the following Stages elements:
1. Work product element of type "work product"
2. Work product element of type "work product state"
3. Activity element of type "activity"
4. Resource element of type "script"

Following figure shows a simple example of a workflow modeled with Stages.

![Figure 5: Workflow Elements](image)

5.3.2 Relations between Workflow Elements

Relation: Work Product State - Work Product

Now that we know the minimal set of Stages elements a workflow needs we have to link these elements within Stages. The association between work product and work product state is a parent-child link. Every work product can have several work product states (children). A work product state must have a parent work product. Figure 6 on page 15 illustrates this relation between a work product and its states.
Relation: Work Product State - Activity

A workflow can be seen as a behavior for a specific event. The behavior is represented by the activity element of the workflow. An event is represented by work product states. Basically output and/or input associations connect these activities with work product states. The workflow consists of two work product states which are belonging to the same parent work product. One of them serves as input, the other one as output state. The chapter Modeling Workflows on page 15 describes the workflow in more detail.

Relation: Scripts

Scripts can be assigned to associations between activities and work product states. Scripts contain instructions for the ALM System and are executed, when the corresponding workflow is executed on the ALM System.

5.3.3 Modeling Workflows

The work products of a workflow represent items on the corresponding ALM system. In Jira, these items are called Issues whereas in RTC, they are known as Work Items. As already mentioned in chapter Concrete Process on page 12 you can model processes, which are able to react on state transitions. The associated script must be assigned to the output association of the corresponding activity. A modeled workflow is shown in the following image.
A workflow has to have two work product states from the same parent work product. Work products refer to items and work product states refer to lifecycle states on the ALM System. With that in mind, these workflows can be described like in the note below.

**Note:** Workflows can be read like:

Every time, a user sets the lifecycle state of an item of type 'Story' from state 'New' to 'In Progress', execute the associated script on the ALM System.

### 5.4 Work Product Attributes

This section describes all relevant attributes a work products has to have, to be used in workflows. Details about workflows can be found in chapter *Workflows* on page 13.

The special attributes of work products, which are used in workflows, relate to information from the ALM Systems. The values provided by these attributes create a logical link between the elements used in Stages and the corresponding items of the ALM System.

The following table shows the attributes of work products which represent items of an ALM System:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Generic Work Item Type   | BOOLEAN  | If true, the linked workflow is executed for every existing item of the ALM System.  
If false, the linked workflow is only executed for items, which were created from this or another workflow but not for items created by the ALM System. |
| Work Item Type           | TEXT     | The link between work products and the corresponding items of the ALM System.  
(e.g. com.ibm.team.apt.workItemType.story for the work item Story in RTC) |

The table below shows the attributes of work product states, which represent the lifecycle states of an item of the ALM System:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Work Item State          | TEXT     | The link between work product states and corresponding lifecycle states of the ALM System.  
(e.g. com.ibm.team.apt.story.verified for the lifecycle state Verified of the work item Story in RTC) |
| Work Item State Group    | BOOLEAN  | If true, the value entered in the text field Work Item State will be interpreted as a group of states. This requires the ALM System to support state groups. |

Another table explains the common attributes of work products and work product states:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes in JSON Format</td>
<td>JSON</td>
<td>Additional information wrapped in a json object which will be accessible during script execution on the ALM System.</td>
</tr>
<tr>
<td>Enactment Description</td>
<td>TEXT</td>
<td>Additional description field, which can be used to fill the description field of items of the ALM System.</td>
</tr>
</tbody>
</table>
For further information on the actual content of these attributes and how to use them for a specific ALM System please read the appropriate modeling guide.