



# ALM Octane

## Git Integration Tool

Software Version: 1.1.2

## Installation Guide

# Introduction

This tool can be used to:

- Fetch pull requests from Git repositories into Octane.
- Get branch information from Git repositories into Octane.
- Create branches from Octane on a Git repository.

**Note:** Only Bitbucket is supported at the moment.

## Limitations

- This solution can be used for one instance of Octane with multiple shared spaces and workspaces. In case you desire to use this utility on more than one Octane instance, you will need to reinstall the utility for each instance.
- In the current version we support only one Bitbucket instance connected to an Octane instance. Multiple repositories and projects from that instance of Bitbucket can be connected to Octane.
- In order to fetch the pull requests and branch information correctly, the commits data must be present in Octane, not only in the repository.
- If the configuration file (**/conf/configuration.properties**) is modified, the server must be restarted.
- After creating a new branch, the branch will be available in Octane only if these three conditions are met:
  - the commits data for that specific branch must be present in Octane
  - the newly created branch stems from a branch that would have already existed in the branch information field
  - branch information is refreshed by pressing the “Get Branch Information” button

If the branch stems from any other branch, it means that it will not contain any of the commits from that branch that exist in Octane and are not already merged in the default branch and thus it will not appear in the branch information.

## Prerequisites

One instance of each application below is required for the integration:

- Octane – Version 12.60.47 and higher
- Bitbucket – Version 5.16 and higher

One of the following web servers:

- Tomcat – Version 7.0 and higher
- Jetty – Version 9.4.20 and higher

## Deployment

### Fill the attributes in the configuration file

In order to install the tool, you will need the **git-integration-for-octane.war** file and a java web application server such as Tomcat/Jetty, where the artifact can be deployed to.

Please follow the steps below after adding the war to your web server (Tomcat/Jetty):

Fill in the configuration file, **configuration.properties**, which can be found in the **/conf** folder.

**Note:** The “\” is a special escape character which can be used in the configuration file to break long lines into multiple lines. Since it is a special character, if you want to use it in a property you have to escape it by using “\\” instead of “\”. [Here](#) you can read more about how this can be used.

**Example:** For the octane.user you can do the following:

```
octane.user= user1, \  
            user2, \  
            user3
```

Key	Value
<b>Octane Fields</b>	
octane.server	<p>The <b>URL</b> to the Octane server.</p> <p>Example: <code>http://octane.company.com:8080</code></p>
octane.sharedSpace	<p>The shared spaces of Octane where the utility will be used. The shared spaces must be separated by a comma.</p> <p>Example: <code>1001,1002</code></p>
octane.user	<p>A username (or API key) which has <b>Space Admin</b> and <b>Workspace Admin</b> permissions. If the utility is used for more than one shared space, you should add a username (or API key) for each shared space separated by a comma. Please pay attention because the order matters.</p> <p>Example: <code>user1,user2</code></p>
octane.password	<p>The password (or Secret) for the user. If there are multiple users listed, all the passwords must be provided in the same order.</p> <p>Example: <code>password1,password2</code></p>
octane.pullRequestsInformationUDFName	<p>In order to display the pull requests, the utility will create a memo field. The name of the field must have “_udf” at the end.</p> <p>Example: <code>pull_requests_udf</code></p>
octane.pullRequestsInformationUDFLabel	<p>The name of the field. This is what users will see in the entities Edit form.</p> <p>Example: <code>Pull Requests</code></p>

octane.branchInformationUDFName	<p>In order to display the branches information, the utility will create a memo field. The name of the field must have “_udf” at the end.</p> <p>Example: branch_information_udf</p>
octane.branchInformationUDFLabel	<p>The name of the field. This is what users will see in the entities Edit form.</p> <p>Example: Branch Information</p>
<b>Bitbucket fields</b>	
repo.host	<p>The only supported value for this field at the moment is <b>bitbucketserver</b>.</p>
bitbucketserver.url	<p>The Bitbucket URL.</p> <p>Example: http://bitbucket.com:7990</p>
bitbucketserver.access	<p>A Bitbucket Personal access token. <a href="#">Here</a> you can find out how to create an access token for Bitbucket.</p>
<b>Proxy fields (Optional)</b>	
proxy.host	<p>The proxy host.</p> <p>Example: webcache.example.com</p>
proxy.port	<p>The proxy port.</p> <p>Example: 8080</p>
<b>Log folder location (Optional)</b>	
logs.location	<p>An existing path for the log files. The folder octane_utility_logs will be added to the input path.</p> <p>Example: C:/example/folder/for/logs</p>

	<p><b>**</b> If this field is not present, the logs will be placed by default in the &lt;git-integration-for-octane&gt;/octane_utility_logs.</p>
--	--

**Note:** Please pay attention to enter the shared spaces and credentials in the correct order!

**Example:** user1 with password1 is Space Admin in the Shared Space 1001 and user2 with password2 is Space Admin in the Shared Space 1002. The correct values can be:

```
octane.sharedSpace=1001,1002
octane.user=user1,user2
octane.password=password1,password2
```

OR

```
octane.sharedSpace=1002,1001
octane.user=user2,user1
octane.password=password2,password1
```

## How to create a button using the External action editor

Please complete the following steps in order to add a button in Octane.

1. Login to Octane with a Site Admin user.
2. Click on the **Settings** icon.
3. Go to **External action editor**. You can find help about this feature [here](#). All the used attributes are explained in the Help Center.
4. Add the JSON in the editor. (There are some JSON examples below)
5. After adding this JSON (with the updated URL) in the External action editor, you need to press **UPDATE** so that Octane will add the button to the entities actions menu.
6. Refresh Octane.

## Add the Get Pull Requests button to Octane

Please read the [“How to create a button using the External action editor”](#) section first.

This example JSON can be used to create the button. The only thing which must be replaced is the highlighted first part of the Tomcat/Jetty URL (for example <http://localhost:9090/git-integration-for-octane>).

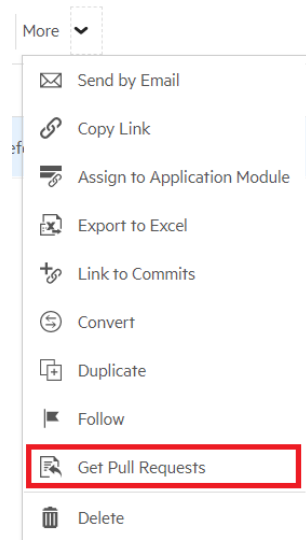


### Example:

```
[ {  
  "name": "pull_requests_button",  
  "title": "Get Pull Requests",  
  "entity_type": ["work_item"],  
  "views": ["list", "details"],  
  "icon": "import",  
  "url": "http://localhost:9090/git-integration-for-octane/pull-  
requests?ids={entity_ids}&sharedSpace={shared_space}&workSpace={workspace}&  
dialogId={dialog_id}&server={octane_url}",  
  "single_entity": false,  
  "events": true,  
  "dialog": "small"  
}]
```

Some of these attributes, i.e. name or title, can be modified. Please follow the instructions from the [Help Center](#) before modifying any of the attributes' values.

The button should be available in the entities action menu if you select at least one entity, on the detailed view, Backlog or Team Backlog views, as below:



### Add the Get Branch Information button to Octane

Please read the [“How to create a button using the External action editor”](#) section first.

The following JSON example can be copied in the External action editor (Do not forget to replace the highlighted first part of the Tomcat/Jetty URL for example **http://localhost:9090/git-integration-for-octane**):

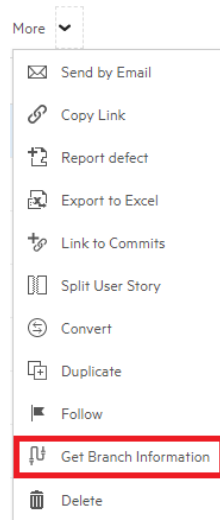


```
Example: [ {  
  "name": "branch_information_button",  
  "title": "Get Branch Information",  
  "entity_type": ["work_item"],  
  "views": [ "list", "details"],  
  "icon": "type-pipeline",  
  "url": "http://localhost:9090/git_integration_for_octane/branch-  
information?ids={entity_ids}&sharedSpace={shared_space}&workSpace={workspac  
e}&dialogId={dialog_id}&server={octane_url}",  
  "single_entity": false,  
  "events": true,  
  "dialog": "small"  
} ]
```

Some of these attributes, i.e. name or title, can be modified. Please follow the instructions from the [Help Center](#) before modifying any of the attributes' values.



The button should be available in the entities action menu if you select at least one entity, on the detailed view, Backlog or Team Backlog views, as below:



### Add the Create Branch button to Octane

Please read the [“How to create a button using the External action editor”](#) section first.

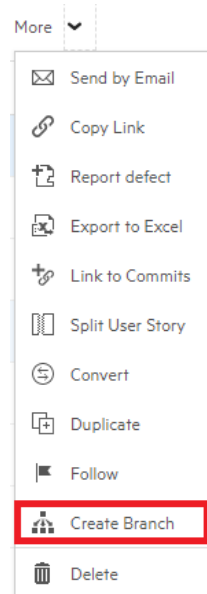
The following JSON example can be copied in the External action editor (Do not forget to replace the highlighted first part of the Tomcat/Jetty URL for example **http://localhost:9090/git-integration-for-octane**):



```
Example: [ {  
  "name": "create_branch_button",  
  "title": "Create Branch",  
  "entity_type": ["work_item"],  
  "views": ["list", "details"],  
  "icon": "children",  
  "url": "http://localhost:9090/git_integration_for_octane/create-branch-  
page?id={entity_id}&sharedSpace={shared_space}&workSpace={workspace}&server  
={octane_url}",  
  "single_entity": true,  
  "events": true  
}]
```

Some of these attributes, i.e. name or title, can be modified. Please follow the instructions from the [Help Center](#) before modifying any of the attributes' values.

The button should be available in the entities action menu if you select at least one entity, on the detailed view, Backlog or Team Backlog views, as below:



**Note:** This button will be available only for a single entity! If you select multiple items in the grid view, the button will not be displayed.

## Upgrade

Please follow the next steps for upgrading from an old version of the utility:

1. Stop Tomcat/Jetty.
2. Backup the **/conf** folder and all of its contents to a different location.
3. Delete the old git-integration-for-octane webapp.
4. Copy the .war file to the Tomcat/Jetty webapps folder (this step is optional).
5. Unpack the .war file to the Tomcat/Jetty webapps folder.
6. In the **<git-integration-for-octane>/conf/configuration.properties** file fill in the necessary information. (Some of the properties can be copied from the **/conf** folder backup)
7. Start Tomcat/Jetty.

## Troubleshooting

By default, the middleware does not display any errors.

The logs of the application are available by default in the <git-integration-for-octane>/octane\_utility\_logs. However, this can be configured, and the location can be different. Please see the configuration file [property](#) for this.

In case of any errors or latencies please check the logs for ERROR and WARNING messages.

## Feedback

This is our [GitHub](#) page. Please feel free to share your feedback and suggestions with us there.

