ALM Octane

Git Integration Tool

Software Version: 1.1.2

User Guide

Document Release Date: October 2019

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

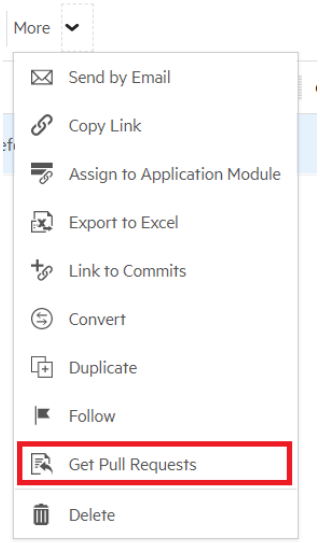
- In order to fetch the pull requests and branch information correctly, the commits data must be present in Octane, not only in the repository.

- 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.

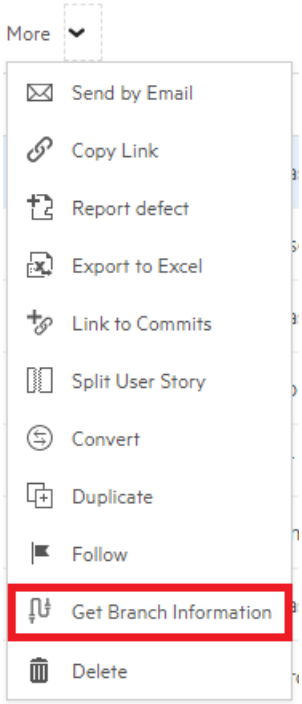
Get Pull Requests button in Octane

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:



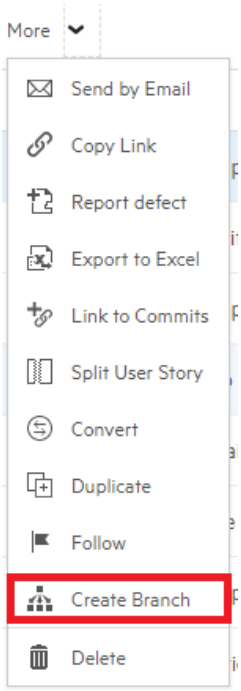
Get Branch Information button in Octane

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:



Create Branch button in Octane

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:

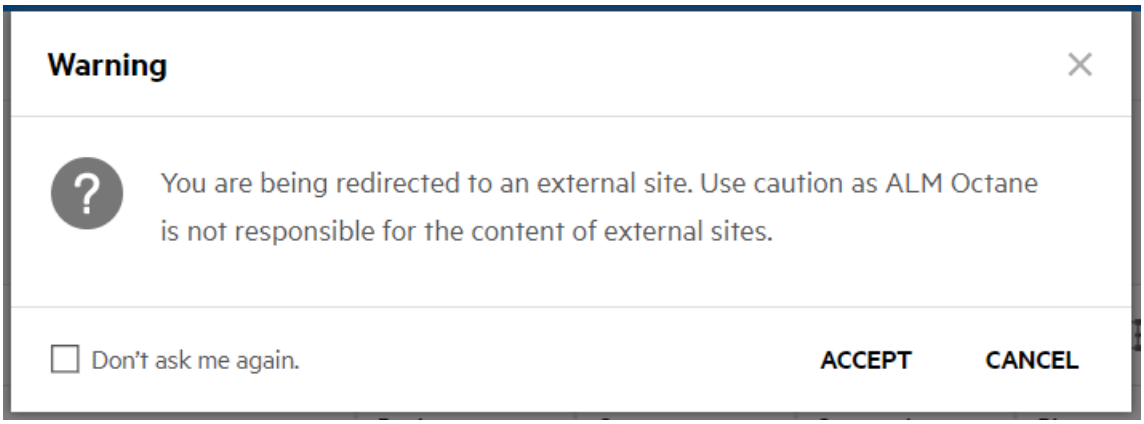


Note: This button will be available only for a single entity! If you select multiple items in the grid view, the button will not be displayed.

First time usage

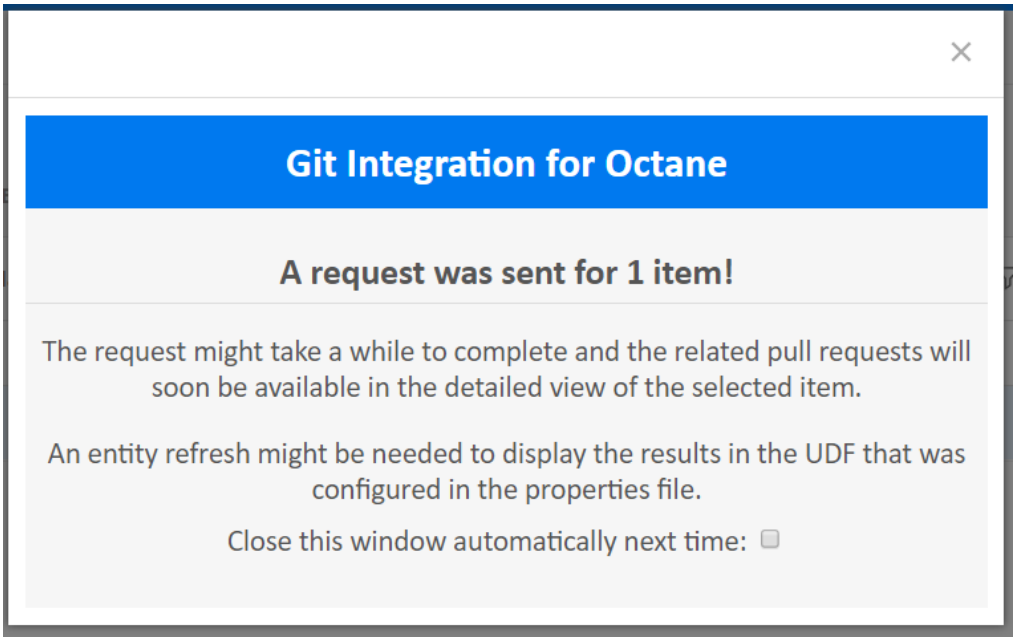
> **Note:** The process of getting branch information is similar to getting the pull requests which is detailed below.

Once you have your commits data from Bitbucket brought into Octane, you can start using the tool. You can select all the entities and press the "**Get Pull Requests**" button. For the first time, Octane will display this message:



This is because the Get Pull Requests button, which was just added to Octane, fires a request to the middleware. If you don't want to see this pop up every time you click the button, you can check the "**Don't ask me again**" checkbox.

For every call (button press), the middleware will display the following content:



If the checkbox is selected, this window will automatically close after 4 seconds and the detailed view of the entities will be refreshed.

> **Note:** The content of the window will not change in case there are any errors or delays. The calls to fetch the pull requests are done while this message is displayed. In case of any latencies please check the logs (\git-integration-for-octane\logs) for errors or warnings.

The middleware will perform the following actions when the button is pressed for the first time:

1. For epics, features, defects, user stories and quality stories, the memo UDF (where the pull requests will be listed) will be created.

2. The UDF will be added into the Edit form for every entity listed above. This means that users will see the UDF when they edit an entity.

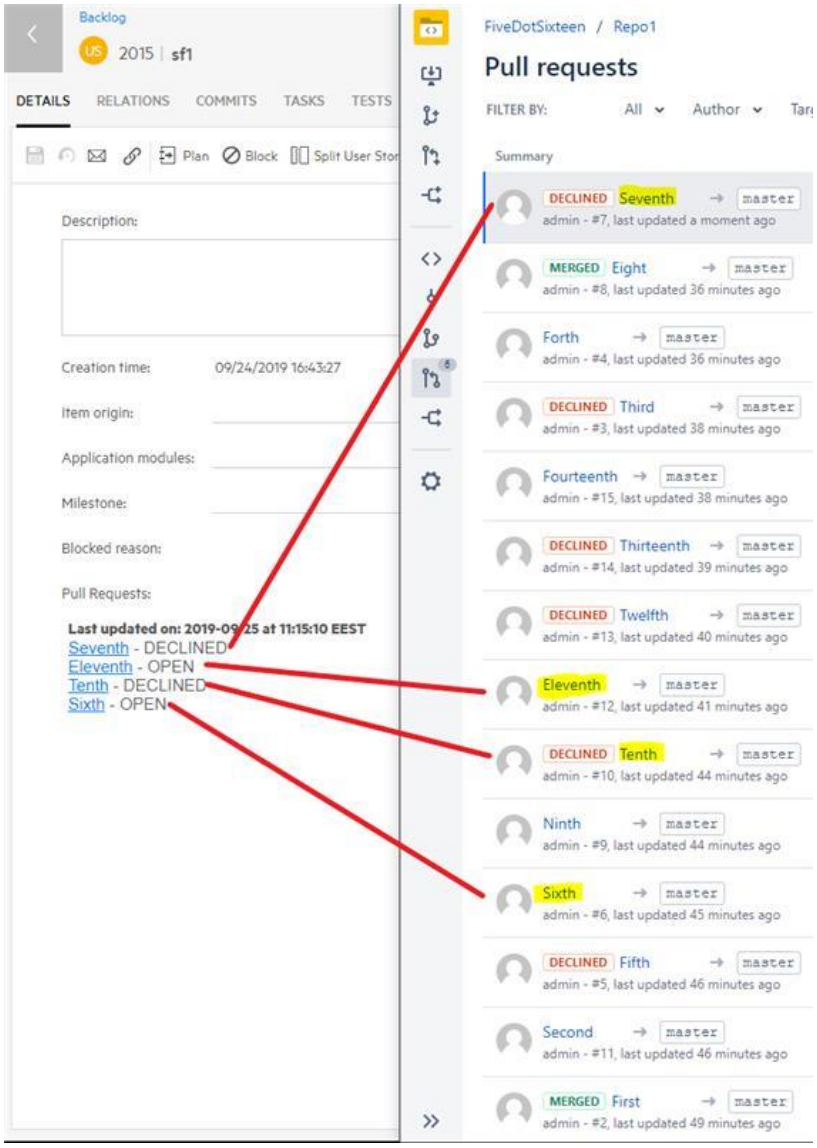
3. A new rule will be added, in order to make the UDF read only. This rule will be created for all the entities mentioned above.

   The rule makes the UDF read-only for all the users, except the ones with Space Admin permissions.

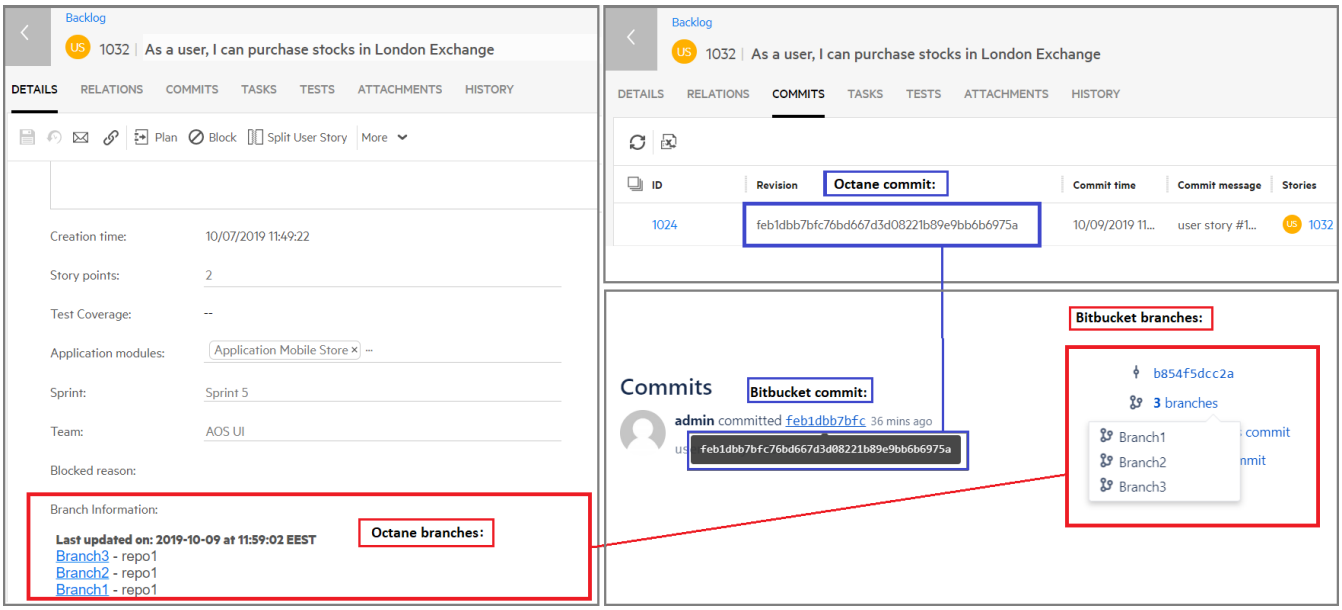
For Isolated Spaces, the items mentioned above will be created at workspace level. For Shared Spaces, the items will be created in the master workspace and inherited in the rest of the workspaces.

> **Note:** The UDFs will be created on the first request which is done on one or multiple entities (defect/user story/epic/feature/quality story), which are linked to at least one commit which is included in a pull request. This happens for the branch information UDFs as well.

After all the operations are completed the pull requests links and states will be entered into the memo UDF. The pull requests are ordered as they are in Bitbucket, last updated first. In the pull request memo UDF we can find the following information: the name of the pull request, a link to Bitbucket pull request and the state of the pull request.

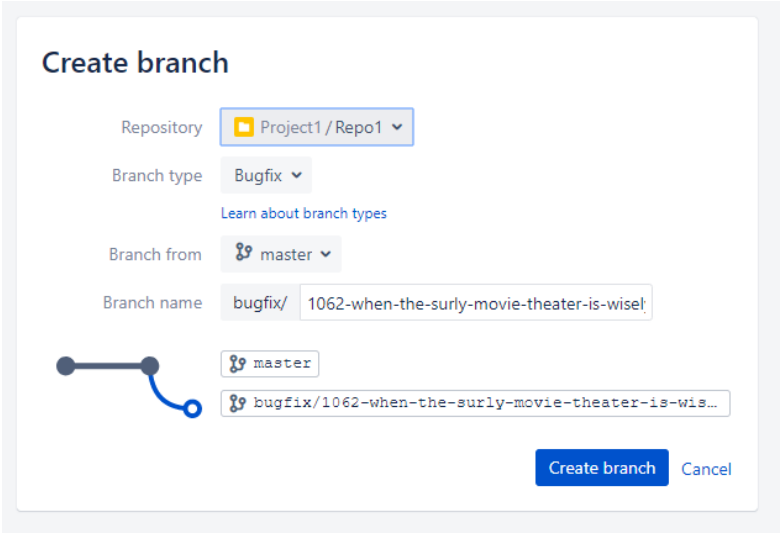


The branches are listed in Octane in a memo UDF, similar to the pull requests. The following information is found there: the name of the branch, a link to Bitbucket for source code on the branch and the repository where the branch has been created.



The commit linked to the user story is present in 3 branches in Bitbucket. The commit can be searched in Bitbucket, where a list with the branches which include the specific commit can be found.

The branch creation redirects to Bitbucket. The name and the type of the branch are automatically selected based on the Octane work items. The branch name is constructed from the work item ID and name.



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** subfolder. However, this can be configured, and the location can be different.

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.