

Unleash Plugin

Plugin Information

View Unleash Maven [on the plugin site](#) for more information.

This is a Jenkins Build Wrapper for Maven Jobs. It enables you to perform Maven releases using the [unleash-maven-plugin](#).

How To Use The Plugin?





This section describes the project configuration options as well as the actual usage of the plugin when building Maven releases.


Project Configuration

In order to use the plugin for building Maven releases you will have to enable it first in the configuration of your Maven build job. Go to the **Build Environment** section and enable the build wrapper by checking the **Unleash** box.

Then you will see some input fields for basic settings. There you can add the required profiles and options for your release build. The goals are pre-defined because currently `unleash:perform` is the only goals that makes sense there.

Unleash

Goals	<input type="text" value="unleash:perform -X"/>	
Profiles	<input type="text"/>	
Additional Release Arguments	<input type="text"/>	
SCM Credentials	<input type="text" value="shillner"/> 	<input type="button" value="Add"/>



Clicking the Advanced... button shows some additional settings for workflow overriding, SCM-related settings and some checkboxes that affect the actual release form that we use later.

Unleash

Goals	<input type="text" value="unleash:perform -X"/>	?
Profiles	<input type="text"/>	?
Additional Release Arguments	<input type="text"/>	?
SCM Credentials	<input type="text" value="shillner"/> <input type="button" value="Add"/>	
Workflow File Path	<input type="text" value="wf/perform"/>	?
Hook Data	<input type="button" value="Add"/>	?
Use Log Timestamps	<input checked="" type="checkbox"/>	?
Preselect 'Specify A Global Version For All Modules'	<input type="checkbox"/>	?
Preselect 'Allow Local Release Artifacts'	<input checked="" type="checkbox"/>	?
Preselect 'Commit Before Tagging'	<input type="checkbox"/>	?
Tag Name Pattern	<input type="text" value="@{project.version}"/>	?
SCM Message Prefix	<input type="text" value="[unleash-maven-plugin]"/>	?














The Hook Data section can then be used for really advanced stuff. If you f.i. extend the default workflow and add an exec-hook to the workflow, you will be able to provide data for the hook execution at this point.

Unleash








Goals	<input type="text" value="unleash:perform -X"/>	?									
Profiles	<input type="text"/>	?									
Additional Release Arguments	<input type="text"/>	?									
SCM Credentials	<input type="text" value="shillner/*****"/> Add										
Workflow File Path	<input type="text" value="wf/perform"/>	?									
Hook Data	<div><div>X</div><table><tr><td>Name</td><td><input type="text" value="exec[1]"/></td><td>?</td></tr><tr><td>Data</td><td><input type="text" value="echo test"/> ▼</td><td>?</td></tr><tr><td>Rollback Data</td><td><input type="text" value="echo rollback"/> ▼</td><td>?</td></tr></table></div> <p>Add</p>	Name	<input type="text" value="exec[1]"/>	?	Data	<input type="text" value="echo test"/> ▼	?	Rollback Data	<input type="text" value="echo rollback"/> ▼	?	?
Name	<input type="text" value="exec[1]"/>	?									
Data	<input type="text" value="echo test"/> ▼	?									
Rollback Data	<input type="text" value="echo rollback"/> ▼	?									
Use Log Timestamps	<input checked="" type="checkbox"/>	?									
Preselect 'Specify A Global Version For All Modules'	<input type="checkbox"/>	?									
Preselect 'Allow Local Release Artifacts'	<input checked="" type="checkbox"/>	?									
Preselect 'Commit Before Tagging'	<input type="checkbox"/>	?									
Tag Name Pattern	<input type="text" value="@{project.version}"/>	?									
SCM Message Prefix	<input type="text" value="[unleash-maven-plugin]"/>	?									

Use The Plugin

The usage of the plugin is pretty simple and is nothing more than submitting a form on a subpage of the project. After you've enabled the plugin a link will be shown on the left side of the screen called **Trigger Unleash Maven Plugin**:

-  [Workspace](#)
-  [Build Now](#)
-  [Delete Maven project](#)
-  [Configure](#)
-  [Modules](#)
-  [Dependency Graph](#)
-  [Subversion Polling Log](#)
-  [Unit Tests \[core\]](#)
-  [Servertest Results](#)
-  [TestNG Results](#)
-  [Perform Maven Release](#)
-  [Trigger Unleash Maven Plugin](#)
-  [Move](#)

-  [U](#)
-  [S](#)
-  [T](#)
-  [W](#)
-  [L](#)
-  [R](#)
-  [L](#)
-  [D](#)

Build History			trend
<input type="text" value="find"/>			
 #10	27.07.2016 11:24	39 MB	
 #9	27.07.2016 11:10	 462 MB	
 #8	27.07.2016 11:05	39 MB	
 #7	27.07.2016 11:01	39 MB	
 #6	27.07.2016 10:54	 39 MB	

-  [Late](#)
-  [Late](#)

After clicking on this link you will be directed to the form page where you can start the release build for your project.

Versioning

- Specify A Global Version For All Modules ?
- Use automatic version calculation for each module ?

Module	Release Version	Development Version
module-1	100	101-SNAPSHOT
release-test-1	100	101-SNAPSHOT

Advanced

- Allow Local Release Artifacts ?
- Commit Before Tagging ?

[Schedule Maven Release Build](#)

On this page you can either specify a global version for all modules of the project (for release and dev) or you stick to the versions provided by the plugin and listed in the table which allows independent versioning of modules. You can also provide custom SCM credentials and request some other stuff.

Versioning

- Specify A Global Version For All Modules ?
Release Version
Development Version
- Use automatic version calculation for each module ?






Advanced

- Allow Local Release Artifacts ?
- Commit Before Tagging ?

[Schedule Maven Release Build](#)

Release Info

Once you have some successful or failed release builds you can also get some project-related release information, such as the badge icons indicating successful or failed releases.

 #82	19.07.2016 10:41	
 #81	19.07.2016 10:34	
 #80	19.07.2016 10:13	

There are also two permalinks available, one for successful and one for failed releases.

Permalinks

- [Last build \(#10\), 1 hr 32 min ago](#)
- [Last stable build \(#10\), 1 hr 32 min ago](#)
- [Last successful build \(#10\), 1 hr 32 min ago](#)
- [Last failed build \(#6\), 2 hr 3 min ago](#)
- [Last unsuccessful build \(#6\), 2 hr 3 min ago](#)
- [Last completed build \(#10\), 1 hr 32 min ago](#)
- [Last Successful Release Build \(#9\), 1 hr 47 min ago](#)
- [Last Failed Release Build \(#6\), 2 hr 3 min ago](#)

Finally there are some ListView columns:

S	Name ↓	Last Successful Release	Last Failed Release
  	kernel-unleash	1 hr 55 min (2.28.44)	2 hr 11 min (2.28.44)

Icon: [S](#) [M](#) [L](#)

[Legend](#)  [RSS for all](#)  [RSS for failures](#)  [RSS for just latest builds](#)

Support

Issues around this plugin are tracked in the Jenkins Issue Tracker. To browse open issues please visit [this page](#). There you can also file new ones if necessary.

General information about the usage of the Maven plugin can be found here: [unleash-maven-plugin](#).

Copyright Information

Substantial parts of the plugin sources have been extracted from the M2 Release Plugin. It served as a base for getting started since the feature set was very similar.
Please find further copyright notes in the source code.

Version History

1.0.0 (July 28, 2016)

The initial version of the plugin.

Unleash Maven Plugin versions:

Minimum required: 1.0.0

Maximum supported:

1.1.0 (August 25, 2016)

Added support for credentials managed by the credentials plugin. These credentials are passed to the Maven plugin as environment variables. The plugin supports Username/Password credentials as well as SSH credentials. In case of SSH credentials only the passphrase is used for the SCM providers. The private key must be present in the .ssh directory.

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

1.2.0 (August 29, 2016)

- Fixed [JENKINS-37752](#): Release profile appending is buggy

- Implemented [JENKINS-37408](#): Add checkboxes for -X and -e on the formular site

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

1.2.1 (August 29, 2016)

The release profiles of the configuration are now also used for the unleash invocation since there was some confusion and some issues when building reactor projects that enable modules depending on profiles (the release build has built the modules which have been enabled by the profiles but the unleash build did not install and deploy them since these modules were not active during the unleash build).

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

2.0.0 (September 29, 2016)

There are no functional or non-functional changes of the plugin sources. The only reason this release has been published was that the licensing of this plugin has been switched. The plugin is now published under MIT license to better fit into the Jenkins ecosystem.

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

2.1.0 (November 02, 2016)

- Implemented [JENKINS-36981](#): keep release builds. The advanced section of the unleash project configuration now contains an input field for the number of successful release builds to lock in the history.

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

2.1.1 (November 15, 2016)

- Implemented [JENKINS-39714](#): No badge shown for unstable builds. A yellow badge icon is now shown for running release builds as well as unstable ones. The latest successful release permalink and the appropriate list view column now also show unstable builds

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

2.2.0 (April 25, 2017)

- Fixed [JENKINS-36983](#): The version calculation for the badge icons as well as the list view columns and the release overview page has been changed thoroughly. It shouldn't happen again that the badges or the columns show unknown versions.
- Added an additional input field for the development version upgrade strategy. For more information about that feature please see [Unleash Maven Plugin Wiki - VersionUpgradeStrategy](#).

Unleash Maven Plugin versions:

Minimum required: 2.1.0

Maximum supported:

2.3.0 (May 11, 2017)

- Fixed [JENKINS-44050](#): Improved SSH-based authentication by passing the private key that was referenced as a credential to the maven plugin. The private key will not be visible in the logs but will be set as an environment variable during the build.

Unleash Maven Plugin versions:

Minimum required: 2.6.0

Maximum supported: