

deploy:deploy

Full name:

org.apache.maven.plugins:maven-deploy-plugin:3.1.4:deploy

Description:

Deploys an artifact to remote repository.

Attributes:

- Requires a Maven project to be executed.
- The goal is thread-safe and supports parallel builds.
- Binds by default to the lifecycle phase (<https://maven.apache.org/ref/current/maven-core/lifecycles.html>) : `deploy`.

Optional Parameters

Name	Type	Since	Description
<code><allowIncompleteProjects></code>	boolean	3.1.1	Set this to <code>true</code> to allow incomplete project processing. By default, such projects are forbidden and Mojo will fail to process them. Incomplete project is a Maven Project that has any other packaging than "pom" and has no main artifact packaged. In the majority of cases, what user really wants here is a project with "pom" packaging and some classified artifact attached (typical example is some assembly being packaged and attached with classifier). Default: <code>false</code> User Property: <code>allowIncompleteProjects</code>

<code><altDeploymentRepository></code>	String	-	Specifies an alternative repository to which the project artifacts should be deployed (other than those specified in <code><distributionManagement></code>). Format: <code>id::url</code> id The id can be used to pick up the correct credentials from the settings.xml url The location of the repository Note: In version 2.x, the format was <code>id::layout::url</code> where <code>layout</code> could be <code>default</code> (ie. Maven 2) or <code>legacy</code> (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout. User Property: <code>altDeploymentRepository</code>
<code><altReleaseDeploymentRepository></code>	String	2.8	The alternative repository to use when the project has a final version. Note: In version 2.x, the format was <code>id::layout::url</code> where <code>layout</code> could be <code>default</code> (ie. Maven 2) or <code>legacy</code> (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout. See also: <code>altDeploymentRepository</code> User Property: <code>altReleaseDeploymentRepository</code>
<code><altSnapshotDeploymentRepository></code>	String	2.8	The alternative repository to use when the project has a snapshot version. Note: In version 2.x, the format was <code>id::layout::url</code> where <code>layout</code> could be <code>default</code> (ie. Maven 2) or <code>legacy</code> (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout. See also: <code>altDeploymentRepository</code> User Property: <code>altSnapshotDeploymentRepository</code>
<code><deployAtEnd></code>	boolean	2.8	Whether every project should be deployed during its own deploy-phase or at the end of the multimodule build. If set to <code>true</code> and the build fails, none of the reactor projects is deployed. Default: <code>false</code> User Property: <code>deployAtEnd</code>

<code><retryFailedDeploymentCount></code>	int	2.7	Parameter used to control how many times a failed deployment will be retried before giving up and failing. If a value outside the range 1-10 is specified it will be pulled to the nearest value within the range 1-10. Default: 1 User Property: <code>retryFailedDeploymentCount</code>
<code><skip></code>	String	2.4	Set this to 'true' to bypass artifact deploy Since since 3.0.0-M2 it's not anymore a real boolean as it can have more than 2 values: <code>true</code> : will skip as usual <code>releases</code> : will skip if current version of the project is a release <code>snapshots</code> : will skip if current version of the project is a snapshot - any other values will be considered as <code>false</code> Default: false User Property: <code>maven.deploy.skip</code>

Parameter Details

`<allowIncompleteProjects>`

Set this to `true` to allow incomplete project processing. By default, such projects are forbidden and Mojo will fail to process them. Incomplete project is a Maven Project that has any other packaging than "pom" and has no main artifact packaged. In the majority of cases, what user really wants here is a project with "pom" packaging and some classified artifact attached (typical example is some assembly being packaged and attached with classifier).

- **Type:** `boolean`
- **Since:** `3.1.1`
- **Required:** No
- **User Property:** `allowIncompleteProjects`
- **Default:** `false`

`<altDeploymentRepository>`

Specifies an alternative repository to which the project artifacts should be deployed (other than those specified in `<distributionManagement>`).

Format: `id::url`

id

The id can be used to pick up the correct credentials from the settings.xml

url

The location of the repository

Note: In version 2.x, the format was `id::layout::url` where `layout` could be `default` (ie. Maven 2) or `legacy` (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout.

- **Type:** `java.lang.String`
 - **Required:** `No`
 - **User Property:** `altDeploymentRepository`
-

<altReleaseDeploymentRepository>

The alternative repository to use when the project has a final version. **Note:** In version 2.x, the format was `id::layout::url` where `layout` could be `default` (ie. Maven 2) or `legacy` (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout.

See also: `altDeploymentRepository`

- **Type:** `java.lang.String`
 - **Since:** `2.8`
 - **Required:** `No`
 - **User Property:** `altReleaseDeploymentRepository`
-

<altSnapshotDeploymentRepository>

The alternative repository to use when the project has a snapshot version. **Note:** In version 2.x, the format was `id::layout::url` where `layout` could be `default` (ie. Maven 2) or `legacy` (ie. Maven 1), but since 3.0.0 the layout part has been removed because Maven 3 only supports Maven 2 repository layout.

See also: `altDeploymentRepository`

- **Type:** `java.lang.String`
 - **Since:** `2.8`
 - **Required:** `No`
 - **User Property:** `altSnapshotDeploymentRepository`
-

<deployAtEnd>

Whether every project should be deployed during its own deploy-phase or at the end of the multimodule build. If set to `true` and the build fails, none of the reactor projects is deployed.

- **Type:** `boolean`
 - **Since:** `2.8`
 - **Required:** `No`
 - **User Property:** `deployAtEnd`
 - **Default:** `false`
-

<retryFailedDeploymentCount>

Parameter used to control how many times a failed deployment will be retried before giving up and failing. If a value outside the range 1-10 is specified it will be pulled to the nearest value within the range 1-10.

- **Type:** `int`
- **Since:** `2.7`
- **Required:** `No`

- **User Property:** `retryFailedDeploymentCount`
 - **Default:** `1`
-

<skip>

Set this to 'true' to bypass artifact deploy Since since 3.0.0-M2 it's not anymore a real boolean as it can have more than 2 values:

- `true` : will skip as usual
 - `releases` : will skip if current version of the project is a release
 - `snapshots` : will skip if current version of the project is a snapshot
 - any other values will be considered as `false`
 - **Type:** `java.lang.String`
 - **Since:** `2.4`
 - **Required:** `No`
 - **User Property:** `maven.deploy.skip`
 - **Default:** `false`
-