Working with Ivy

Overview

Artifactory fully supports working with Ivy both as a source for artifacts needed for a build, and as a target to deploy artifacts generated in the build process.

For Ivy to work with Artifactory, the following files must be present and configured:

1. The Ivy settings file: `ivysettings.xml` is used to configure resolution and deployment of artifacts using repositories in Artifactory.
2. The Ivy modules file: `ivy.xml` is where the project's modules and dependencies are declared.
3. The Ant build file: `build.xml` is used to execute the ANT tasks that will, in turn, use Ivy for resolution and deployment of artifacts.

Page Contents

Ivy Settings - `ivysettings.xml`

The `ivysettings.xml` file holds a chain of Ivy resolvers for both regular artifacts and Ivy module files. These are used to resolve and publish (i.e. deploy) artifacts.

There are a two ways to configure resolvers in *`ivysettings.xml`* in order to set up Ivy to work with Artifactory:

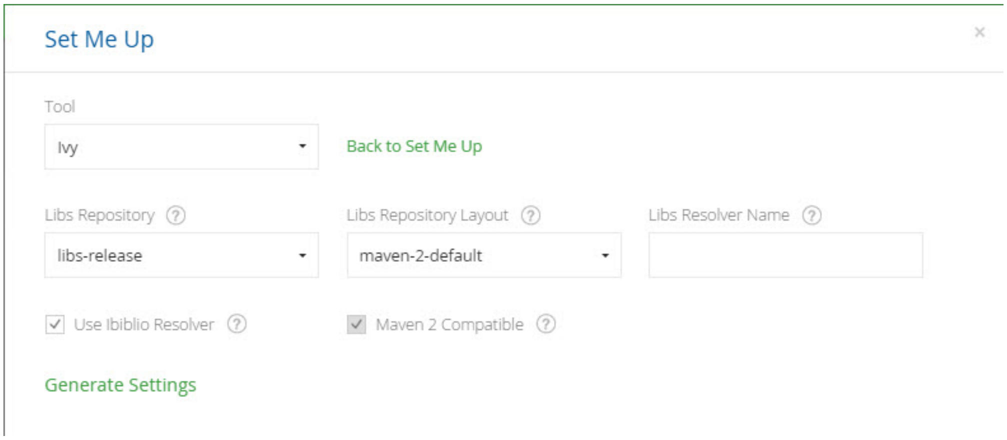
1. Automatically, using the Artifactory Ivy Settings Generator
2. Manually defining IBiblio and URL resolvers.

Automatic Settings with Artifactory's Ivy Settings Generator

To begin quickly, you can define credentials and resolver settings using Artifactory's Ivy Settings Generator. This generates a URL resolver suitable for resolution.

In the **Artifact Repository Browser** of the **Artifacts** module, select **Set Me Up.** In the **Set Me Up** dialog, set **Ivy** in the **Tool** field and click "Generate Ivy Settings". You can now specify the repositories you want to configure for Ivy.

Since the **Libs Repository** field only includes virtual or remote repositories, none of these will be suitable for deployment, and you need to modify the deployment URL to point to a local repository.


Set Me Up

Tool: Ivy

Libs Repository: libs-release

Libs Repository Layout: maven-2-default

Libs Resolver Name:

☑ Use Ibiblio Resolver

☑ Maven 2 Compatible

Choose an Ivy Repository Layout

Be sure to select layout that is compatible with Ivy such as **ivy-default** or a custom layout that you have defined.

Provisioning Dynamic Settings for Users

You can deploy and provision a dynamic settings template for your users.

Once downloaded, settings are generated according to your own logic, and can automatically include user authentication information.

For details, please refer to Provisioning Build Tool Settings under Filtered Resources.

Defining a Manual Resolver

The IBiblio Resolver

This resolver is only used to resolve dependencies. By default, it assumes artifacts in your repository are laid-out in the popular and standard Maven 2 format (which may not always be the case).

The IBiblio resolver can resolve artifacts from remote Maven 2 HTTP repositories, and if you use version ranges it relies on `maven-metadata.xml` files in the remote repository to gather information on the available versions.

To use the IBiblio resolver, add the following to your `ivysettings.xml file`:

```
<resolvers>
    <ibiblio name="artifactory" m2compatible="true" root="http://localhost:8080/artifactory/libs-releases"/>
</resolvers>
```

The URL specified in the `root` property must point to an Artifactory repository. In the above example, it is the pre-configured `libs-releases` virtual repository.

The `m2compatible` property configures the resolver with an artifact pattern that follows the standard Maven 2 layout.

The URL Resolver

The URL resolver can be used to resolve dependencies and/or for deployment of both regular artifacts and Ivy module files.

To publish or resolve artifacts to or from Artifactory, you need to configure a URL resolver with the pattern that matches your target repository layout for both Ivy and artifact files.

For example:

```
<!-- Authentication required for publishing (deployment). 'Artifactory Realm' is the realm used by
Artifactory so don't change it. -->
<credentials host="localhost" realm="Artifactory Realm" username="admin" passwd="password"/>
<resolvers>
    <url name="artifactory-publish">
        <!-- You can use  m2compatible="true" instead of specifying your own pattern -->
        <artifact pattern=
          "http://localhost:8080/artifactory/ivy-local/[organization]/[module]/[revision]/[artifact]-
[revision].[ext]"/>
        <ivy pattern="http://localhost:8080/artifactory/ivy-local/[organization]/[module]/[revision]/ivy-
[revision].xml" />
    </url>
</resolvers>
```

The URL resolver uses HTML href analysis to learn about the available versions of a remote artifact. This is less reliable than using an IBiblio resolver, however it works well with remote Artifactory servers.

Using a Chain Resolver

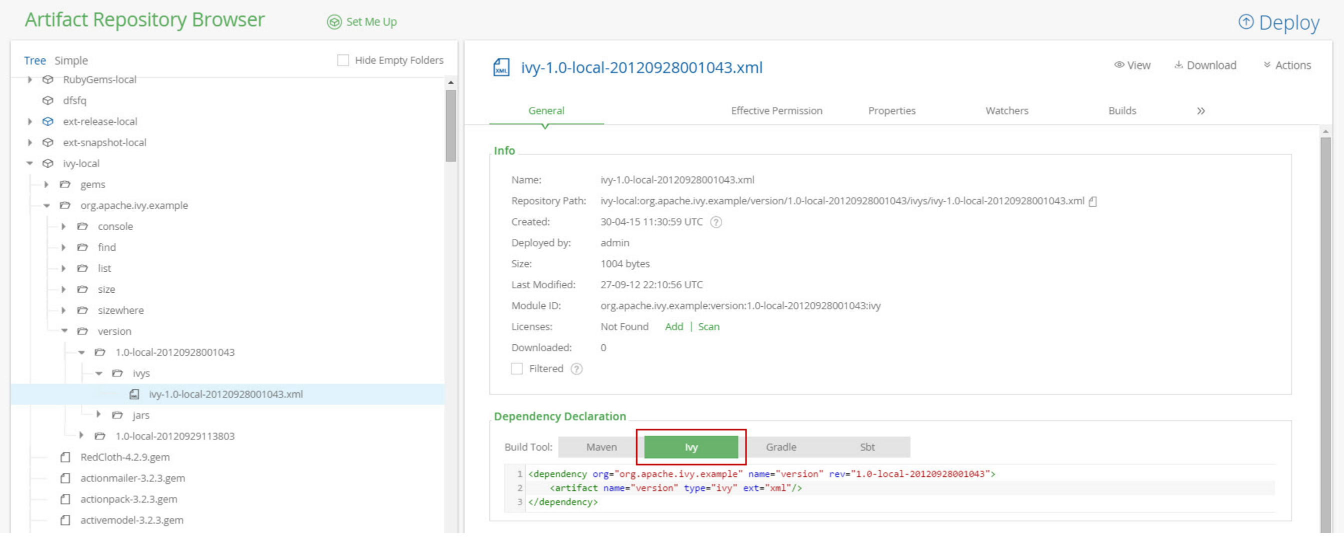
You can combine resolver definitions under a chain resolver in Ivy which uses a set of sub resolvers to resolve dependencies and for publishing.

For details please refer to the Ivy documentation for Chain Resolver.

Ivy Modules - `ivy.xml`

`ivy.xml` files contain a list of dependency declarations that must be resolved for the build.

In the **Artifact Repository Browser** of the **Artifacts** module, you can obtain dependency declaration snippets by selecting either an Ivy module, or a POM artifact, and copying the Ivy **Dependency Declaration** section into your `ivy.xml` file.



ivy-1.0-local-20120928001043.xml

Info

Name: ivy-1.0-local-20120928001043.xml
Repository Path: ivy-local:org.apache.ivy.example/version/1.0-local-20120928001043/ivys/ivy-1.0-local-20120928001043.xml
Created: 30-04-15 11:30:59 UTC
Deployed by: admin
Size: 1004 bytes
Last Modified: 27-09-12 22:10:56 UTC
Module ID: org.apache.ivy.example:version:1.0-local-20120928001043:ivy
Licenses: Not Found Add | Scan
Downloaded: 0
□ Filtered

Dependency Declaration

Build Tool: Maven [✓ Ivy] Gradle Sbt

```
<dependency org="org.apache.ivy.example" name="version" rev="1.0-local-20120928001043">
    <artifact name="version" type="ivy" ext="xml"/>
</dependency>
```

Ant Build - `build.xml`

To work with Ivy to resolve dependencies, you need to use <`ivy:configure/`> in your `build.xml` file. This will load the Ivy settings from `ivysettings.xml`.

Artifacts are resolved using <`ivy:retrieve/`>.

For details please refer to the Ivy documentation for Ant Tasks.

Publishing to Artifactory

You can use the <`ivy:publish`> command to configure Ivy to deploy your artifacts into Artifactory using the specified resolver.

For example:

```
<ivy:publish resolver="artifactory-publish" overwrite="true">
   <!--
   Use overwrite="true" if you wish to overwrite existing artifacts
   and publishivy="false" if you only want to publish artifacts not module descriptors
   -->
   <artifacts/>
</ivy:publish>
```

Using a Dedicated Settings File for Deployment

If you have specified deployment settings with the required credentials in a dedicated settings file, you can refer to them by assigning a unique ID.

For example, the following code snippet assigns the deployment settings with the id `ivy.publish.settings`:

```
<ivy:settings id="ivy.pub.settings" file="publish_to_artifactory_settings.xml"/>
```

Then, the publishing task points to these settings using the following attribute in the `publish` element:

```
settingsRef="ivy.pub.settings"
```

For details please refer to the Ivy documentation for Ant Tasks.