

P2 Repositories

Overview

Artifactory provides advanced support for proxying and caching of P2 repositories and aggregating P2 metadata using an Artifactory virtual repository which serves as a single point of distribution (single URL) for Eclipse, [Tycho](#) and any other P2 clients.

This virtual repository aggregates P2 metadata and P2 artifacts from underlying repositories in Artifactory (both local and remote) providing you with full visibility of the P2 artifact sources and allowing powerful management of caching and security for P2 content.

For more information on defining virtual repositories please refer to [Virtual Repositories](#).



For P2 support we recommend using Eclipse Helios (version 3.6) and above.

Older versions of Eclipse may not work correctly with Artifactory P2 repositories.

Page Contents

- [Overview](#)
- [Configuration](#)
 - [Defining a Virtual Repository](#)
 - [Selecting Local Repositories](#)
 - [Selecting Remote Repositories](#)
 - [Creating the Repositories](#)
 - [Eclipse](#)
- [Integration with Tycho Plugins](#)
- [Multiple Remote Repositories with the Same Base URL](#)
- [Configuring a Remote Repository for GWT](#)

Integration Benefits

[JFrog Artifactory and Eclipse P2 Repositories](#)

Configuration

To use P2 repositories, follow the steps below:

- [Define a virtual repository](#)
- [Select local repositories to add to your virtual repository](#)
- [Select remote repositories to add to your virtual repository](#)
- [Create the selected local and remote repositories in your virtual repository](#)
- [Configure Eclipse to work with your virtual repository](#)

Defining a Virtual Repository

- From the **Administration** module, select **Repositories | Repositories | Virtual** and Create a new virtual repository and set **P2** as the **Package Type**.

New Virtual Repository

Basic

Package Type *



P2

Repository Key *

p2-virtual



If developers in your organization use different versions of Eclipse (e.g. Helios and Juno), we recommend that you define a different P2 virtual repository for each Eclipse version in use.

Selecting Local Repositories

Adding a local repository to your virtual P2 repository does not require any special configuration:

- Simply select the desired local repository from the **Local Repository** field. Usually, this will be either a Maven or a Generic repository.
- In the **Path Prefix** field, specify the path to the P2 metadata files (`content.jar`, `artifacts.jar`, `compositeContent.xml` etc.). If left empty, the default is to assume that the P2 metadata files are directly in the repository root directory.
- Click **Add**.

If you have a Tycho repository deployed to a local repository as a single archive, specify the archive's root path. For example: `eclipse-repository.zip!`

Local P2 Repositories

Local Repository

Repo-1.x-2

Path Suffix ?

path/to/metadata

Add

Selecting Remote Repositories

To add a remote P2 repository to Artifactory, enter the URL to the corresponding P2 metadata files (`content.jar`, `artifacts.jar`, `compositeContent.xml`, etc.) and click the "Add" button

A common example is the main Juno repository: <http://download.eclipse.org/releases/juno>

Remote P2 Repositories

P2 Repository URL

Add

Artifactory analyzes the added URL and identifies which remote repositories should be created in Artifactory based on the remote P2 metadata (since remote P2 repositories may aggregate information from different hosts).



When P2 metadata files reside inside an archived file, simply add '!' to the end of the URL.

For example: [http://eclipse.org/equinox-sdk.zip/!](http://eclipse.org/equinox-sdk.zip/)

Creating the Repositories

Once you have selected the local and remote repositories to include in your virtual repository, Artifactory will indicate what action will be taken once you select the **Save & Finish** button.

The possible actions are as follows:

Create*	Creates a new, P2 enabled, remote repository with the given key (you may still edit the remote repository key).
Modify*	Enables P2 support in an existing remote repository.
Include	Adds the repository to the list of repositories aggregated by this virtual repository.
Included	No action will be taken. This repository is already included in the virtual repository.

*For remote repositories only

Filter by Repository

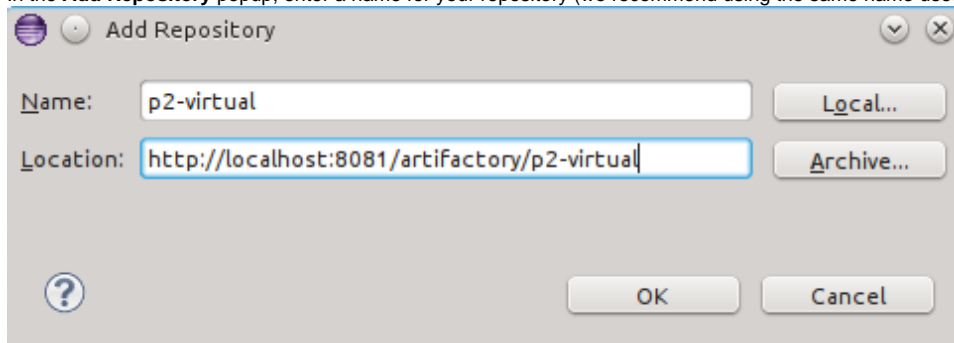
< page 1 of 1 >

Action	Repository	URL
include	download.eclipse.org	http://download.eclipse.org
include	p2-local	local://p2-local/path/to/metadata
include	p2-tycho-local	local://p2-tycho-local/eclipse-repositor...

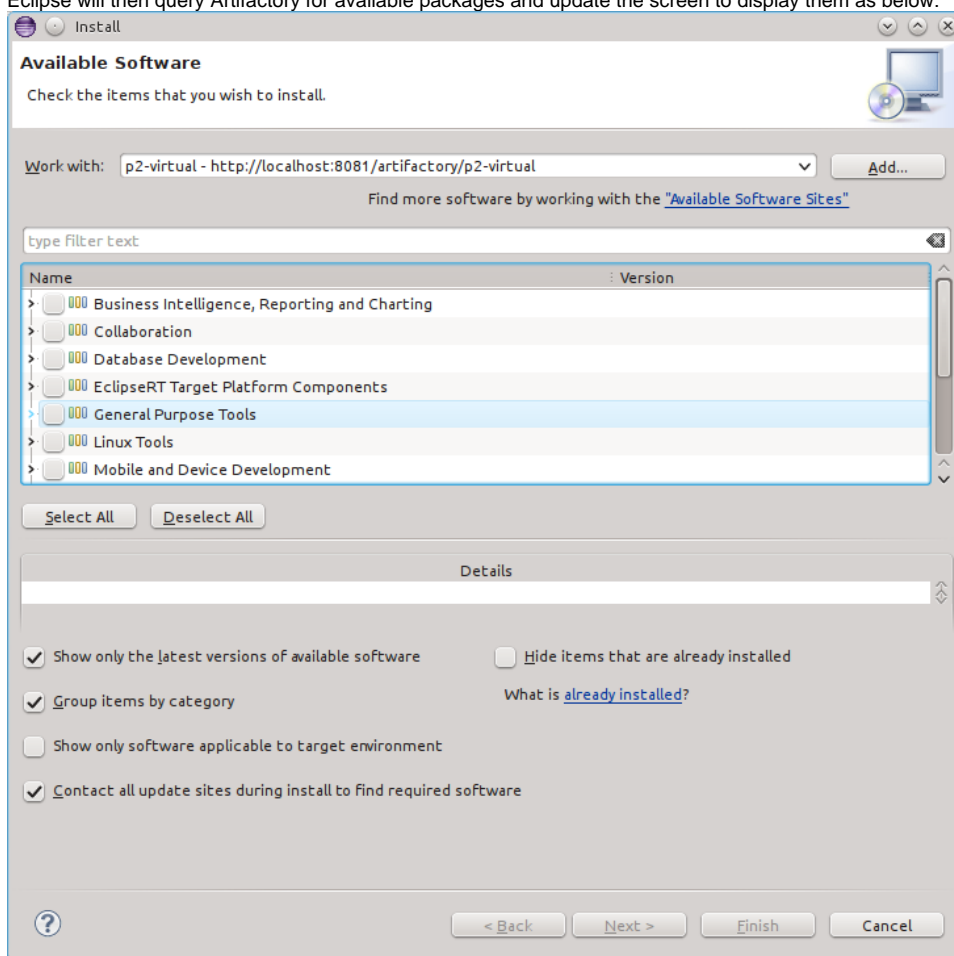
Eclipse

You are now ready to configure eclipse to work with the virtual repository you have created above.

1. In the Eclipse menu, select **Help | Install new Software** and then click **Add**.
2. In the **Add Repository** popup, enter a name for your repository (we recommend using the same name used in Artifactory) and its URL:



3. Eclipse will then query Artifactory for available packages and update the screen to display them as below:



Integration with Tycho Plugins

Artifactory fully supports hosting of Tycho plugins as well as resolving Tycho build dependencies.

To resolve all build dependencies through Artifactory, simply change the repository URL tag of your build's *pom.xml* file as displayed in the snippet below:

```
<repository>
  <id>eclipse-indigo</id>
  <layout>p2</layout>
  <url>http://localhost:8081/artifactory/p2-virtual</url>
</repository>
```



The P2 virtual repository should contain URLs to all local repositories with an optional sub-path in them where Tycho build artifacts reside.

Multiple Remote Repositories with the Same Base URL

When using P2-enabled repositories with multiple remote repositories that have the same base URL (e.g. <http://download.eclipse.org>), you need to ensure that only 1 remote repository is created within your virtual repository (for each base URL). When creating your virtual repository, Artifactory takes care of this for you, but if you are creating the remote repositories manually, you must ensure to create only a single remote repository, and point the sub-paths accordingly in the P2 virtual repository definition.

In the example below, <http://download.eclipse.org/releases/helios> and <http://download.eclipse.org/releases/juno> were both added to the same virtual repository...repository.

Action	Repository	URL
included	download.eclipse.org	http://download.eclipse.org/releases/helios
included	download.eclipse.org	http://download.eclipse.org/releases/juno

...but in fact, the virtual repository only really includes one remote repository

Virtual Repositories

Repository key	Type	Included Repositories	Selected Repositories
p2-virtual-multiremote	P2	1	download.eclipse.org
plugins-release	Maven	3	plugins-release-local ; ext-release-local ; remote-repos
plugins-snapshot	Maven	3	plugins-snapshot-local ; ext-snapshot-local ; remote-repos
remote-repos	Maven	3	java.net.m1 ; repo1 ; gradle-libs

Configuring a Remote Repository for GWT

The Google Plugin for Eclipse does not support Eclipse 4.7 (Oxygen) or later and is no longer available. For more details please refer to the [Google Cloud Platform documentation](#).

To configure your P2 repository to proxy GWT, set the remote repository URL in Artifactory to be: <http://storage.googleapis.com/gwt-eclipse-plugin/v3/release>