

Browsing and Manipulating Maven Repositories

1. Opening the Maven Repository View

To browse Maven repositories and to manipulate repository indexes open the Maven Repositories view by selecting Windows, Show View, Other... as shown in [Figure 6.7, “Opening a View in Eclipse”](#).

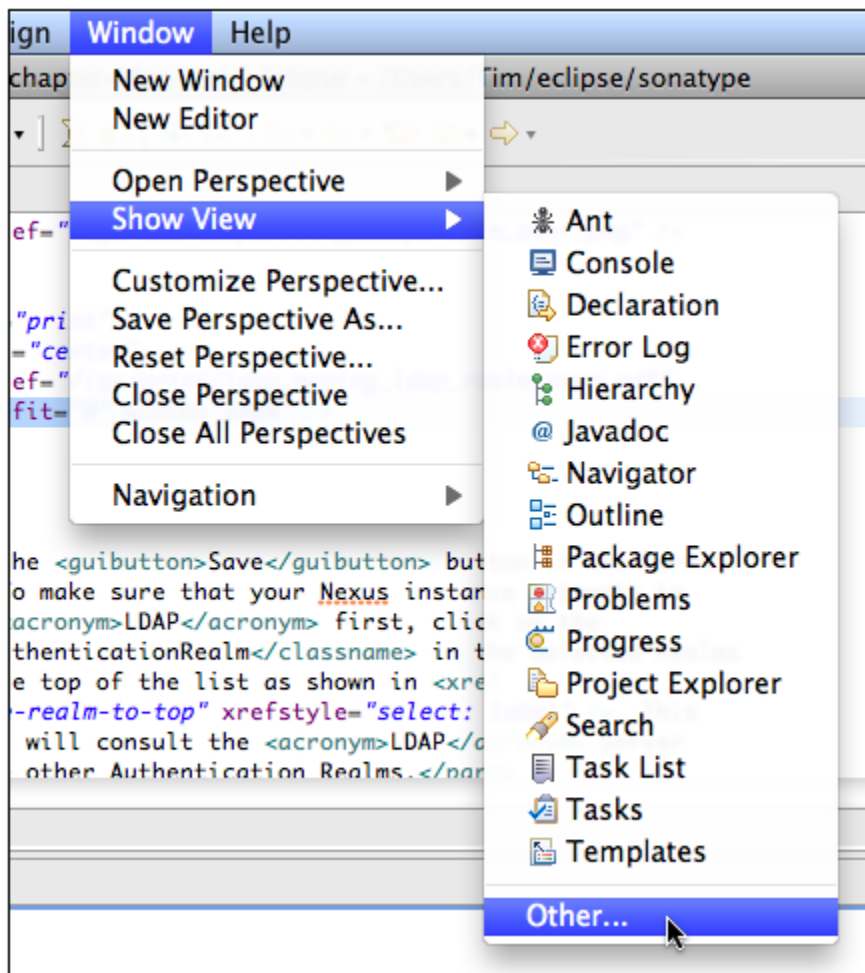


Figure 6.7. Opening a View in Eclipse

Once you select Other... Eclipse will display a dialog containing all available views. Select the Maven Repositories view under the Maven folder in the Show View dialog as shown in [Figure 6.8, “Selecting the Maven Repositories View in the Show View Dialog”](#).

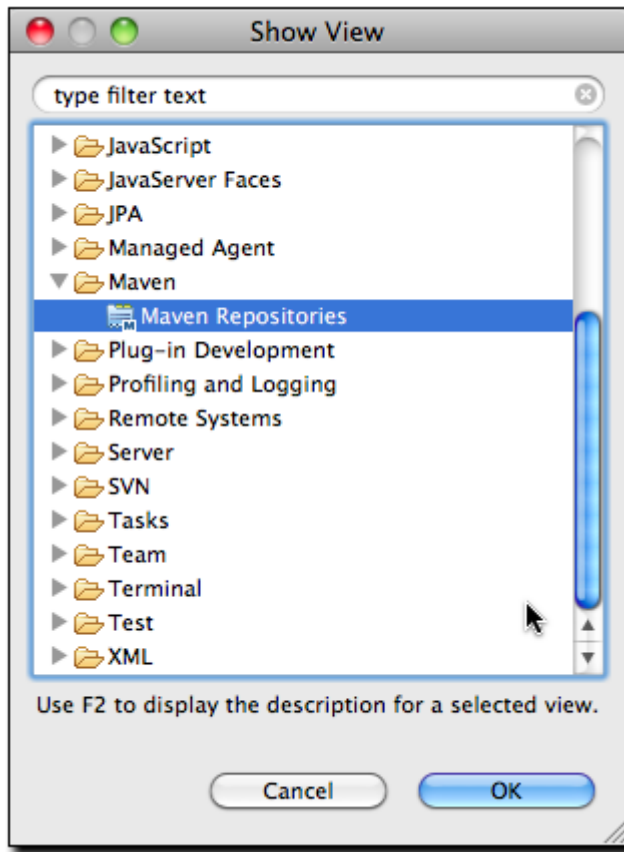


Figure 6.8. Selecting the Maven Repositories View in the Show View Dialog

Once you have selected Maven Repositories and clicked on the OK button, Eclipse will then load the Maven Repositories view. This view contains three folders:

Local Repositories

This folder contains your local Maven repository which is stored in `~/.m2/repository` by default. It also contains a repository that represents the Maven projects contained in your Eclipse workspace.

Global Repositories

This folder contains any global Maven repositories that are referenced by all Maven projects. This folder contains the Central Maven repository under the repository identifier of "central". It will also contain mirrors that have been configured in your Maven Settings (`~/.m2/settings.xml`).

Project Repositories

This folder contains repositories which are defined by your projects. These repositories are present either in your project's `pom.xml` file or in an active Maven Profile.

2. Browsing Global Repositories

If you have been using Maven, you are familiar with the Central Maven repository. This is default repository from which Maven will retrieve dependencies and other artifacts needed during a build. If you expand the central repository, you will be able to browse the contents of the repository and double

click on specific artifacts. Double-clicking on one of the artifacts shown in [Figure 6.9, “Browsing a Global Repository”](#) will load that artifact's POM in the Form-based POM Editor.

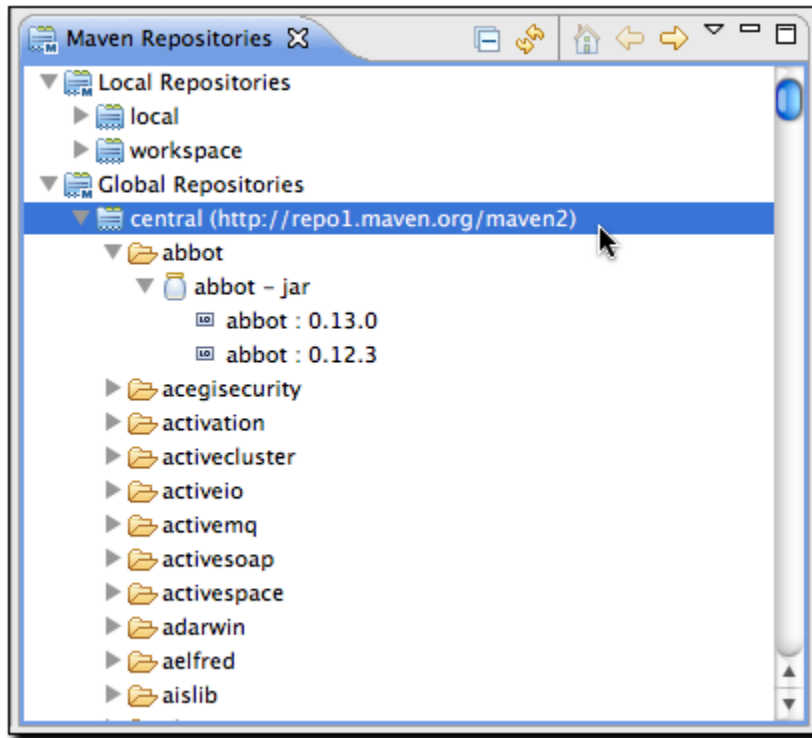


Figure 6.9. Browsing a Global Repository

In addition to loading an artifact's POM in the Form-based POM Editor, you can also right click on a artifact and choose Materialize Project. If the POM for a particular artifact contains valid SCM information, m2eclipse can "materialize" the project from source control into your workspace.

3. Browsing Your Workspace Repository

m2eclipse maintains an index of artifacts generated by your Eclipse workspace. This "workspace" repository is shown in [Figure 6.10, “Browsing the m2eclipse Workspace Repository”](#) under the Local Repository folder. If you expand this folder, you will see artifacts that correspond to your workspace project as shown in [Figure 6.10, “Browsing the m2eclipse Workspace Repository”](#).

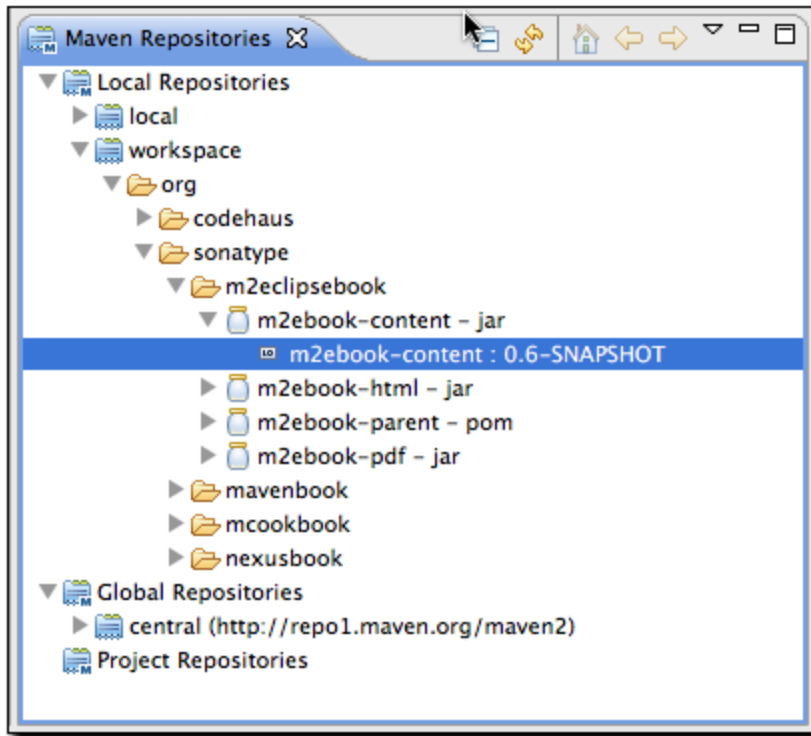


Figure 6.10. Browsing the m2eclipse Workspace Repository

4. Browsing a Project Repository

The Maven Repositories view is also intelligent enough to keep track of any repositories that have been added to your project via your Maven Settings, an active Maven Profile, or that have been added directly to a project's POM. To demonstrate this feature, add a repository element to a pom.xml, by loading the Form-based POM Editor and clicking on the Repositories tab. Click on the "Create..." button and add a new repository with the following values as shown in [Figure 6.11, "Adding a Repository to a Project's POM"](#).

- Repository Identifier: flexmojos
- Repository Name: Flexmojos Repository
- URL: <http://repository.sonatype.org/content/groups/flexgroup/>

Note

You will only see the Repositories tab in your Form-based POM Editor, if you have set your Maven preferences to "Show advanced tabs in the POM Editor" under Eclipse, Preferences..., Maven, POM Editor.

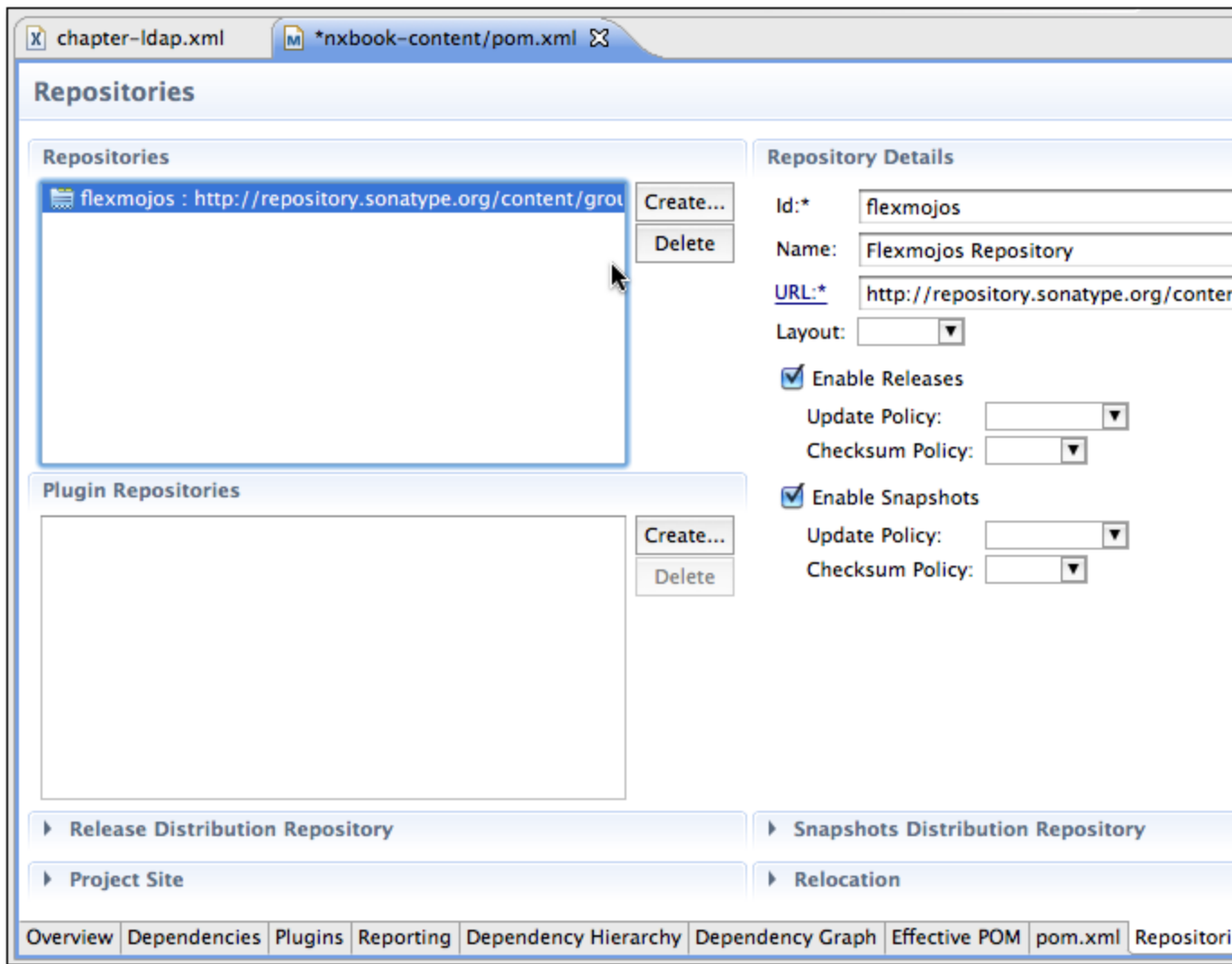


Figure 6.11. Adding a Repository to a Project's POM

Save the POM and open the `pom.xml` tab in the POM Editor. The project's `pom.xml` should contain the repositories element shown in [Figure 6.12, “Project POM with a Custom Repository”](#).

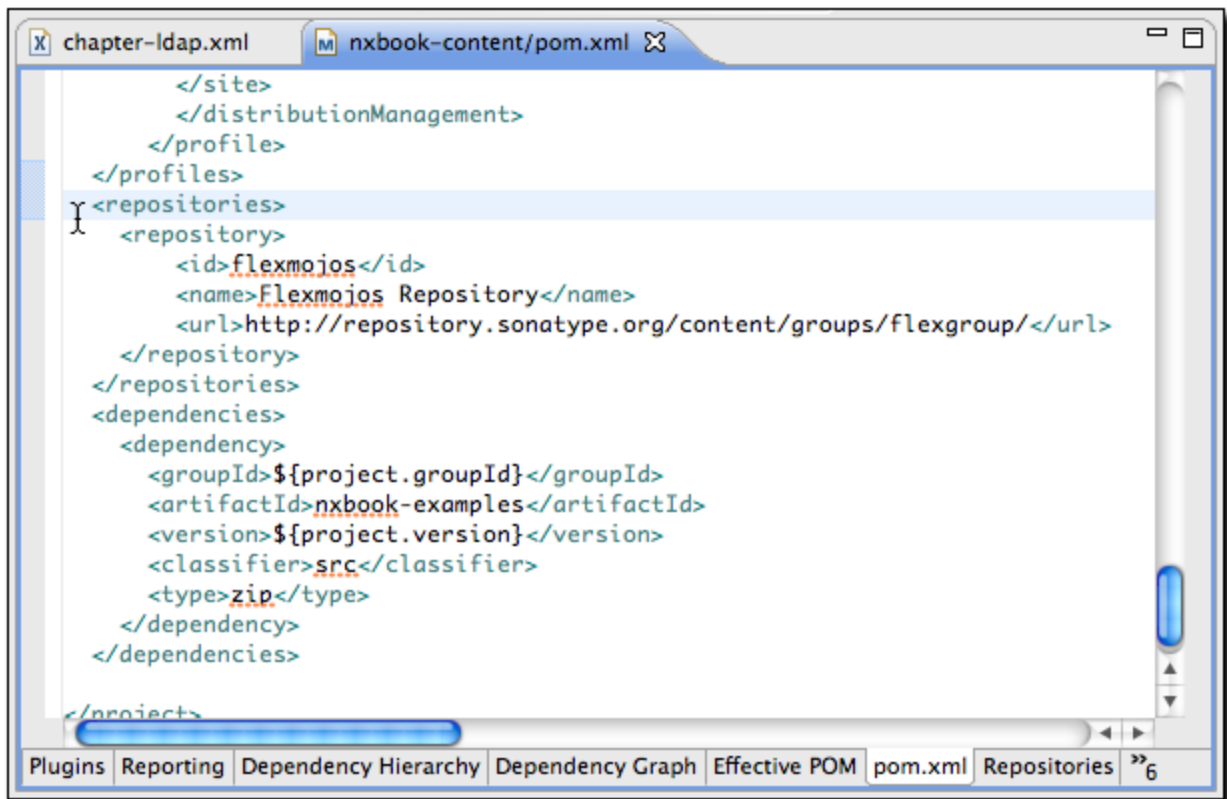


Figure 6.12. Project POM with a Custom Repository

Now that the `pom.xml` contains a custom repository, click on the refresh icon shown in the upper right-hand of [Figure 6.13, “Browsing a Project Repository”](#). The refresh icon looks like two opposing yellow arrows, and clicking this icon will cause the Maven Repositories view to refresh the list of repositories from the selected project and your configured Maven settings.

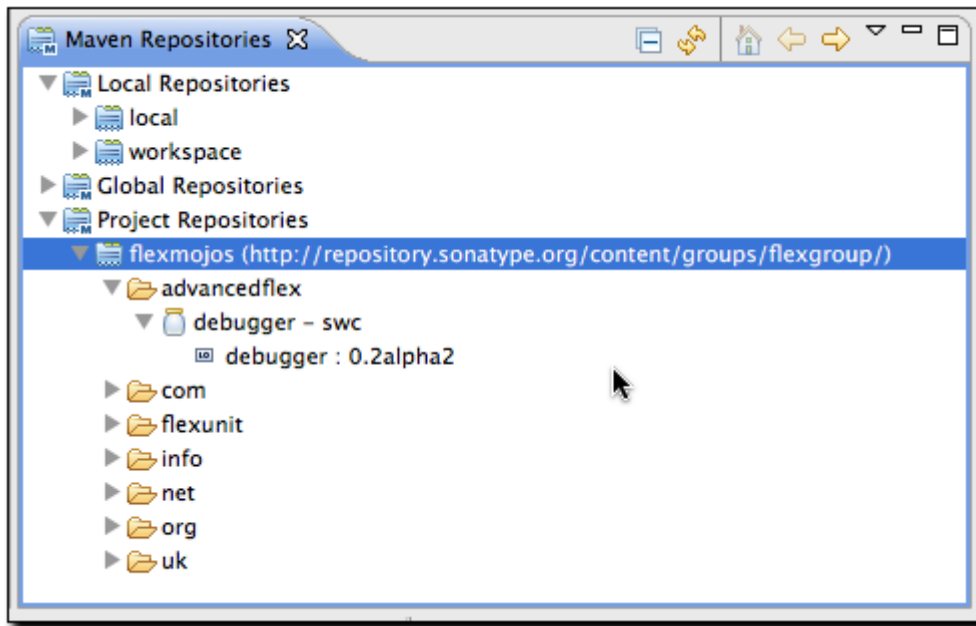


Figure 6.13. Browsing a Project Repository

Once you have added a project repository and clicked on the refresh icon in the Maven Repositories view, you will be able to view the project-specific repository and manipulate the repository index for this project-specific repository.

5. Browsing Your Local Repository

The Maven Repositories view allows you to browse and manipulate your local Maven repository index. m2eclipse maintains an index for the contents of your local repository, you can use this interface to browse artifacts that have been loaded into your local repository as shown in [Figure 6.14, “Browsing Your Local Maven Repository”](#).

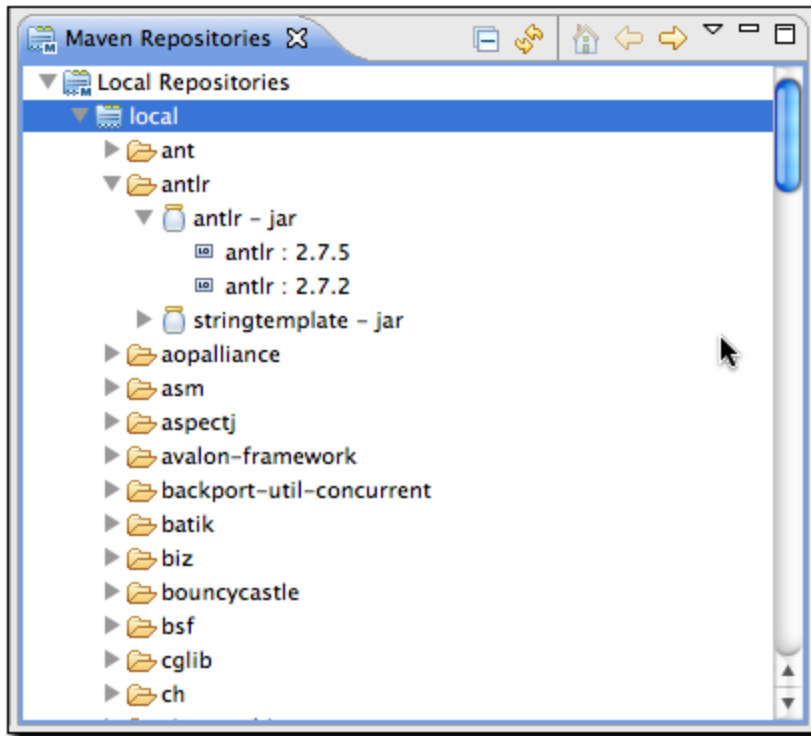


Figure 6.14. Browsing Your Local Maven Repository

6. Manipulating a Repository Index

Every repository that m2eclipse uses is indexed by the Nexus Indexer. If m2eclipse is using a remote repository, it will download a Nexus index from the remote repository. If m2eclipse is managing a local repository (local or workspace) it will use the open source Nexus indexer to create and maintain a local index. This index is what allows you to quickly search for and locate dependencies by artifactId, groupId, version, or classname. You can manipulate the index that is associated with a Maven repository by right-clicking on a repository in the Maven Repositories view and selecting one of the following actions:

Update Index

This will update the index by running an incremental update or by downloading and index from a remote repository.

Rebuild Index

This will rebuild an index for a local repository by iterating through the contents of a repository and recreating a Nexus index from scratch. This can be a useful tool if there is another process outside of Eclipse that is going to be modifying a local Maven repository.

Disable Index

Choosing this option causes m2eclipse to skip index generation for a repository. This can come in handy if you have a series of repositories which you do not want to include in artifact searches. If your organization maintains a number of specialized, segregated repositories that hold snapshots,

you may not want to include these artifacts in simple searches for artifacts that contain a particular identifier or class.

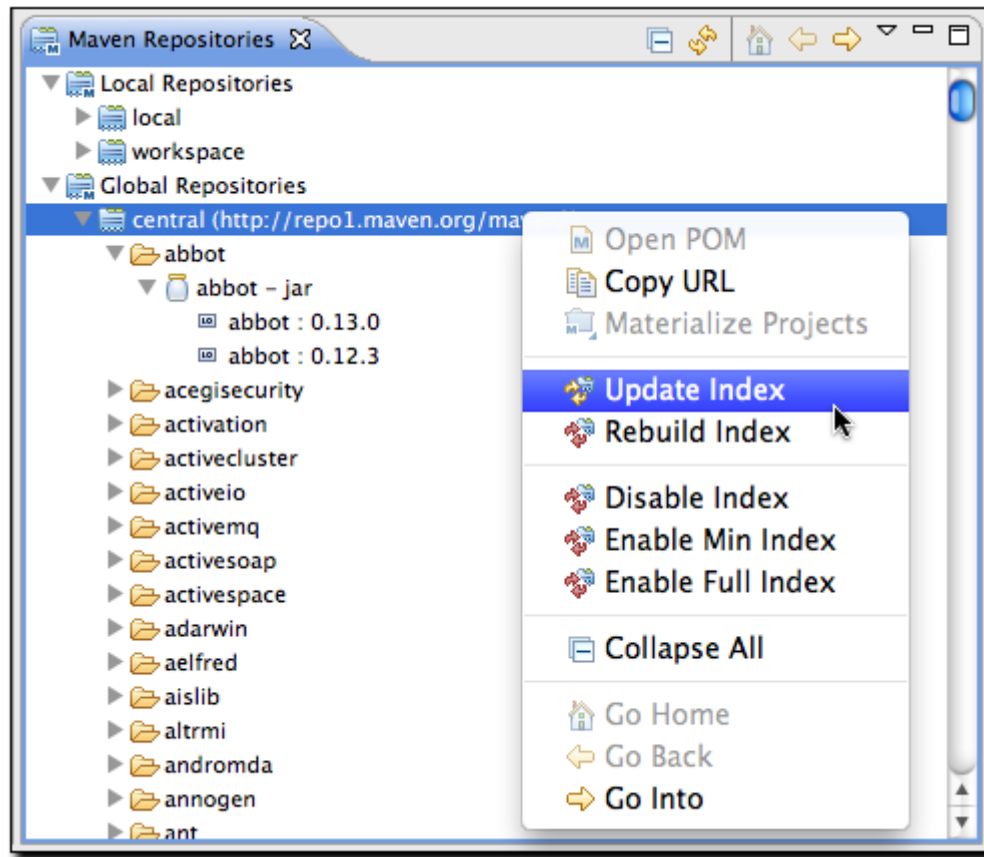


Figure 6.15. Updating a Repository Index

[Figure 6.15, “Updating a Repository Index”](#) shows two indexing options that control the scope of a particular Nexus index:

Enable Min Index

Configures the Nexus Indexer to maintain a minimal index that doesn't contain information about class names.

Enable Full Index

Configures the Nexus Indexer to maintain a full Index that includes the class names contained within each artifact.