



Overtarget Documentation

Version 1.3.0, 2022-03-17T15:02:42Z

Table of Contents

1. Purpose	1
2. Requirements	2
3. Creating a tmodel file	3
3.1. Writing a tmodel file	3
3.2. Extending/Importing a tmodel file	3
4. Generating a target file	4
4.1. Saving the tmodel file	4
4.2. Using the button/context menu "Generate Target"	4
5. Setting a target file as active target	5
6. Other useful stuff	6
6.1. Solving unresolved references in a tmodel	6
6.2. Updating versions	7
7. Your ideas to improve Overtarget	8

Chapter 1. Purpose

Overtarget is a feature that is supposed to help you with the creation and handling of target platform definitions in eclipse.

If you have an eclipse project which is based on other projects, you most likely have similar contents in the target definition files. When you are changing dependencies in the base project, you often have to change them in both files.

Overtarget lets you define your dependencies in a textual `.tmodel` file. Every change is directly compiled into a standard `.target` file. Similar to inheritance of classes you can create another `.tmodel` file for some other project and derive from the previous one. This way you only have to specify the changes, such as added, changed or removed dependencies. All other dependencies are inherited and compiled into the local `.target` file.

This documentation should help you getting familiar with Overtarget and its functionalities.

Chapter 2. Requirements

Overtarget is based on Java / Eclipse and provides an installable feature with plug-ins for your personal eclipse IDE. The following infrastructure is required:

- Java Development Kit (JDK) 8
- Eclipse Oxygen or newer
 - Including Checkstyle
 - Including Spotbugs
 - Including M2E
- Maven 3
- Windows 7 or Linux Computer

Chapter 3. Creating a tmodel file

In an existing project create a new file and add the file extension ".tmodel".

3.1. Writing a tmodel file

A tmodel file starts with the keyword "Target" followed by the name of the tmodel. Within the curly brackets you can define your tmodel the way you like. When a repository location should use all of its units, the keyword "addAll" should be used. A target file is automatically generated when saving the tmodel file.

```
1 Target virsat {
2   OperatingSystem win32
3   WorkingSystem win32
4   Architecture x86_64
5   Locale de_DE
6   TargetJRE JavaSE-1.8
7
8   RepositoryLocation Nebula url "https://download.eclipse.org/nebula/releases/2.2.0/"
9   Unit org.eclipse.nebula.widgets.gallery.feature.feature.group version 1.0.0.201907151344
10  Unit org.eclipse.nebula.widgets.tablecombo.feature.feature.group version 1.2.0.201907151344
11
12 }
13
14 RepositoryLocation EclipseOrbit url "https://download.eclipse.org/tools/orbit/downloads/drops/R2019060221107/repository/"
15 Unit org.antlr.runtime version 3.2.0.v2018111130
16 Unit org.apache.commons.io version 2.6.0.v20190123-2029
17 Unit org.apache.commons.io.source version 2.6.0.v20190123-2029
18
19 // Purely named Eclipse. The name is important for target models having this one as
20 // super model. Then this name can be referenced to explicitly exclude
21 // certain features from the target model
22 RepositoryLocation Eclipse url "http://download.eclipse.org/releases/2019-06/"
23 Unit org.eclipse.emf.compare.ide.ui.feature.group version 3.3.7.201905300200
24 Unit org.eclipse.emf.edapt.runtime.feature.group version 1.4.0.201902190757
25 Unit org.eclipse.emf.sdk.feature.group version 2.10.0.v20190528-0045
26 Unit org.eclipse.emf.transaction.sdk.feature.group version 1.12.0.201805140024
27 Unit org.eclipse.equinox.executable.feature.group version 3.8.400.v20190515-0925
28 Unit org.eclipse.graphiti.sdk.feature.group version 0.16.0.201906041401
29 Unit org.eclipse.jdt.source.feature.group version 5.10.0.v20190605-1800
30 Unit org.eclipse.ocf.all.sdk.feature.group version 5.9.0.v20190603-1146
31 Unit org.eclipse.pde.source.feature.group version 3.14.0.v20190605-1800
32 Unit org.eclipse.platform.sdk version 4.12.0.I20190605-1800
33 Unit org.eclipse.swtbot.eclipse.feature.group version 2.8.0.201906121535
34 Unit org.eclipse.swtbot.eclipse.gef.feature.group version 2.8.0.201906121535
35 Unit org.eclipse.swtbot.feature.group version 2.8.0.201906121535
36 Unit org.eclipse.swtbot.forms.feature.group version 2.8.0.201906121535
37 Unit org.eclipse.swtbot.generator.feature.group version 2.8.0.201906121535
38 Unit org.eclipse.swtbot.ide.feature.group version 2.8.0.201906121535
39 Unit org.eclipse.xtext.sdk.feature.group version 2.18.0.v20190528-0716
40 Unit org.eclipse.gef.feature.group version 3.11.0.201806061308
41 Unit org.eclipse.git.feature.group version 5.4.0.201906121030-rj
42
43 }
44
45 RepositoryLocation Subversion url "https://download.eclipse.org/technology/subversion/4.0/update-site/"
46 Unit org.eclipse.team.svn.feature.group version 4.0.5.I20170425-1700
47
48 RepositoryLocation SubversionConnector url "http://community.polarion.com/projects/subversion/download/eclipse/6.0/update-site/"
49 Unit org.polarion.eclipse.team.svn.connector.feature.group version 6.0.4.I20161211-1700
50 Unit org.polarion.eclipse.team.svn.connector.source.feature.group version 6.0.4.I20161211-1700
51 Unit org.polarion.eclipse.team.svn.connector.sources.feature.group version 6.0.4.I20161211-1700
52 Unit org.polarion.eclipse.team.svn.connector.svnkit18.feature.group version 6.0.4.I20161211-1700
53 Unit org.polarion.eclipse.team.svn.connector.svnkit18.sources.feature.group version 6.0.4.I20161211-1700
54
55 }
```

Example tmodel of virsat



Using content assist (ctrl + spacebar) will suggest the next possible keywords/steps.

3.2. Extending/Importing a tmodel file

It is possible to extend a tmodel by using the keyword "extends". That is especially useful for downstream projects which use the same dependencies as their main project in a tmodel. When there are changes in the tmodel of the main project, the tmodel of the downstream project which extends the main tmodel obtains these changes automatically. A parent target hands down all of its content to the inheriting target, including operating system, windowing system and so on.

The same principle is applicable when importing a tmodel, using the keyword "Import". Imported targets only pass repository locations on to the inheriting targets.

Chapter 4. Generating a target file

4.1. Saving the tmodel file

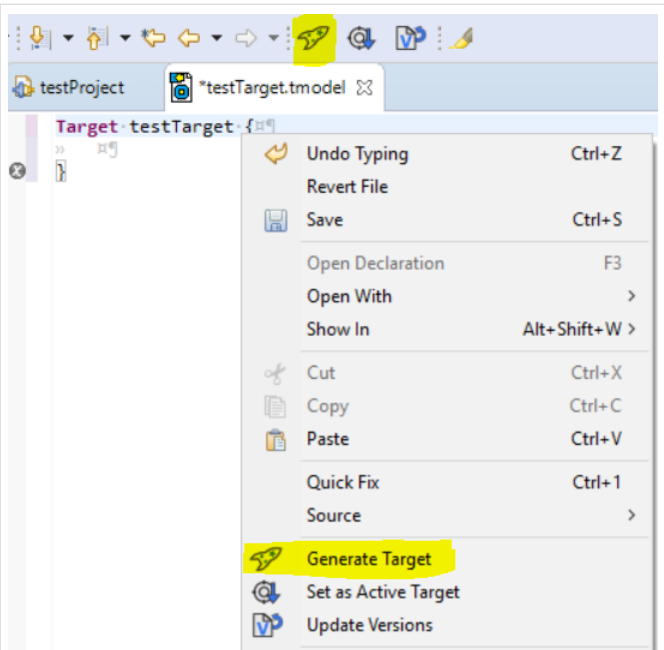
A target file can be generated in different ways. The easiest way would be by saving the tmodel file. The Overtarget generator then automatically generates a target file.



Generating a target file by saving the tmodel file only works when changes have been applied.

4.2. Using the button/context menu "Generate Target"

Another option to generate a target is using the button "Generate Target" in the editor toolbar or in the context menu. A target file will be generated, also when no changes have been applied to the tmodel file.

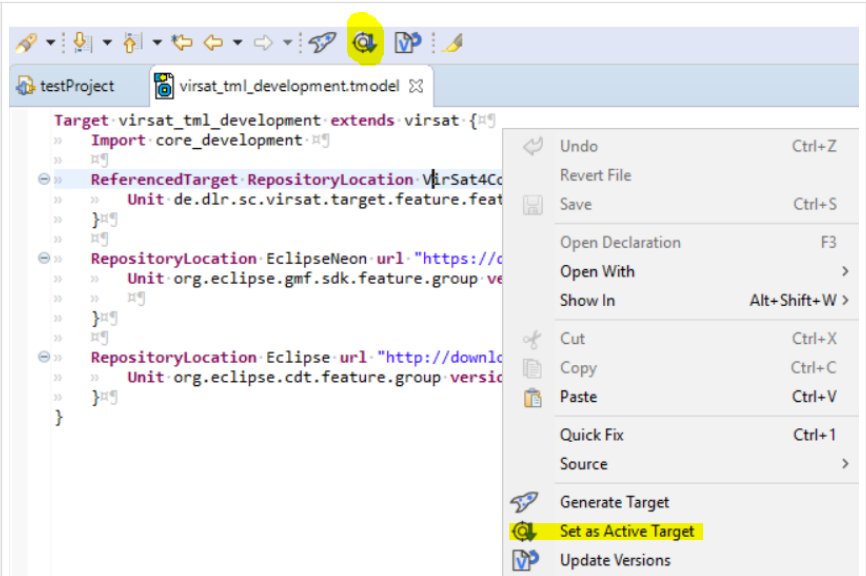


Options to generate a target (highlighted)

Chapter 5. Setting a target file as active target

There are different ways to set a target as active target in eclipse. The usual way is to open the target file which opens the eclipse target editor. Here you can click on "Set as Active Target Platform".

However, Overtarget serves you with a much more comfortable option to set a target. If you want to set your target directly from the tmodel editor you can use the button "Set as Active Target" in the editor toolbar or in the context menu. The target will be automatically loaded and set as the active target platform.



Set as Active Target


Chapter 6. Other useful stuff

6.1. Solving unresolved references in a tmodel

Sometimes when extending or importing another target, Eclipse cannot always resolve the references and shows errors in the tmodel file. It is not possible to generate a target with unresolved references. Overtarget solves the problem by generating a targetForReferences. This target includes the repository locations which lead to the unresolved references. By setting this target as active target, the references are then resolved. In order to generate a targetForReferences, the repositoryLocation to the referenced target has to be added with the keyword "ReferencedTarget". Overtarget offers a quick fix to create a placeholder for a ReferencedTarget.

<p>There is a quick fix to create a placeholder for a ReferencedTarget. After executing the quick fix, the name and url of the repositoryLocation and the unit which leads to the unresolved references.</p>	<div><pre>Target virsat_cef_integration extends virsat_windows { Import core_integration }</pre><div>Couldn't resolve reference to TargetModel 'virsat_windows'. 6 quick fixes available: Create placeholder for referencedTarget Use temporary target to resolve tmodel references Change to 'newTarget' Change to 'virsat_cef_development' Change to 'virsat_cef_integration' Change to 'virsat_cef_release'</div><pre>ReferencedTarget RepositoryLocation <placeholder:virsat> url "<location>" { // add necessary Units here; }</pre></div> <p><i>Quick fix to create placeholder for ReferencedTarget</i></p>
<p>There is a quick fix which suggests to generate a targetForReferences. Executing that quick fix will generate a targetForReferences. The target will also be set as an active target to resolve all references.</p>	<div><div>testProject testTarget.tmodel 11 virsat_tml_development.tmodel referencedTargetTarget</div><div>Target: testTarget extends virsat { ... }</div><div>Couldn't resolve reference to TargetModel 'virsat'. 2 quick fixes available: Generate ReferencedTarget Change to testTarget</div><pre>ReferencedTarget RepositoryLocation VirSat4CoreDevelopment url "https://sourceforge.net/projects/virtualsatellite/files/VirSat4CoreDevelopment/" { Unit de.dlr.sc.virsat.target.Feature.Feature-group version newest; }</pre></div> <p><i>Generate targetForReferences</i></p>

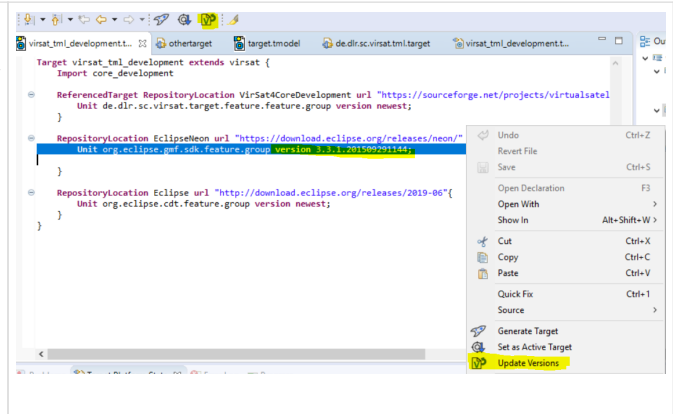
Now there should no longer be problems with unresolved references in the actual target. It can now be generated without complications and then be set as the active target platform.



If there are still unresolved references, please check if you need to add dependencies to the manifest file.

6.2. Updating versions

Another nice feature Overtarget provides, is updating the versions of units in a repository location automatically in the tmodel file. To do that, you have to select the entire line of a unit with its version and then either click the button "Update Versions" in the toolbar or use the context menu. If there is a newer version available, it will be updated.



Chapter 7. Your ideas to improve Overtarget

Overtarget is still in development. So if you have any ideas, how to improve the handling with target platforms, please let us know.

To contribute to this project follow the given steps:

- Create your own fork of the project.
- Apply your changes.
- Create a pull-request of your change to our development branch.

To increase the chance that we accept your pull-request, make sure all tests are working. The best indicator is the Travis CI job. Next we will review your pull-request, give comments and maybe accept it.