



# JBoss Tattletale 1.1 User's Guide

Betraying all your project's naughty little secrets

Copyright © 2009 Red Hat Middleware

---

# Table of Contents

1. About JBoss Tattletale .....	1
1.1. The team .....	1
1.2. Thanks to .....	1
2. Introduction .....	2
3. Getting started .....	3
3.1. Installation .....	3
3.1.1. Maven .....	3
3.1.2. Source code .....	4
3.2. Configuration .....	4
3.2.1. Filtering .....	8
3.3. Running .....	8
4. Apache Ant .....	10
4.1. report .....	10
5. Apache Maven .....	15
5.1. report .....	15
6. Reports .....	20
6.1. Dependency reports .....	20
6.1.1. Dependants .....	20
6.1.2. Depends On report .....	20
6.1.3. Graphical dependencies report .....	21
6.1.4. Transitive Dependants .....	21
6.1.5. Transitive Depends On report .....	21
6.1.6. Circular Dependency report .....	22
6.2. General reports .....	22
6.2.1. Class Location .....	22
6.2.2. OSGi .....	23
6.2.3. Sealed information .....	23
6.2.4. Signing information .....	24
6.2.5. Eliminate Jar files with different versions .....	24
6.2.6. Invalid version .....	25
6.2.7. Multiple Jar files .....	25
6.2.8. Multiple Jar files (Package) .....	26
6.2.9. Multiple Locations .....	26
6.2.10. Unused Jar .....	27
6.2.11. Black listed .....	27
6.2.12. No version .....	28
6.3. Archive reports .....	28
6.3.1. Java ARchive (JAR) .....	28
7. Troubleshooting .....	30
7.1. JBoss Tattletale generates empty reports .....	30
7.2. JBoss Tattletale throws an OutOfMemoryException .....	30
7.3. How do I ? .....	30

## About JBoss Tattletale

JBoss Tattletale is a tool that can help development teams getting an overview of the project they are working on or a product they depend on.

The tool generates reports that will show dependencies and general information that can help identify areas that needs attention such as minimizing the number of dependencies or eliminate duplicated class files from the class path.

JBoss Tattletale will help to improve the quality of your software project.

### 1.1. The team

Jesper Pedersen acts as the lead for the JBoss Tattletale project. He can be reached at `jesper (dot) pedersen (at) jboss (dot) org`.

### 1.2. Thanks to

Jay Balunas, Torben Jaeger, Steve Taranto and James Cobb.

## Introduction

Have you ever found yourself frustrated with a `ClassNotFoundException`? Would you like to know what libraries are in your project and what they depend on? Would you like to get a full report on this stuff every time you run your ant build? If so you need to use the JBoss Tattletale project!

JBoss Tattletale is a tool that can help you get an overview of the project you are working on or a product that you depend on.

The tool will provide you with reports that can help you

- Identify dependencies between JAR files
- Find missing classes from the classpath
- Spot if a class/package is located in multiple JAR files
- Spot if the same JAR file is located in multiple locations
- With a list of what each JAR file requires and provides
- Verify the `SerialVersionUID` of a class
- Find similar JAR files that have different version numbers
- Find JAR files without a version number
- Find unused JAR files
- Identify sealed and signed JAR archives
- Locate a class in a JAR file
- Get the OSGi status of your project

JBoss Tattletale will recursive scan the directory pass as the argument for JAR files and then build the reports as HTML files.

JBoss Tattletale is licensed under GNU Lesser General Public License (LGPL) version 2.1 or later.

We hope that JBoss Tattletale will help you in your development tasks !

Please, visit the official JBoss Tattletale project page at <http://www.jboss.org/tattletale/>.

## Getting started

### 3.1. Installation

JBoss Tattletale can be downloaded in its binary form for easy installation.

The download location is: <http://www.jboss.org/tattletale/downloads>

Once downloaded extract the files by executing:

```
unzip jboss-tattletale-1.1.0.GA.zip
```

or

```
tar xzf jboss-tattletale-1.1.0.GA.tar.gz
```

depending on which archive type you downloaded.

JBoss Tattletale is now located in a folder under the directory you extracted the files into.

#### 3.1.1. Maven

The JBoss Tattletale project is published in the JBoss Maven2 repository:

```
repository.jboss.org
```

under the group id of: `org.jboss.tattletale`

The artifacts for the project are

- `tattletale`: The core library
- `tattletale-ant`: The Apache Ant tasks
- `tattletale-maven`: The Apache Maven plugin

Developer snapshots are published on the JBoss Snapshots Maven2 server:

```
snapshots.jboss.org
```

### 3.1.2. Source code

If you want to experiment with the latest developments you may checkout the latest code from SVN. Be aware that the information provided in this manual might then not be accurate.

The anonymous SVN repository is located under:

```
svn co http://anonsvn.jboss.org/repos/tattletale/trunk/ tattletale-trunk
```

The developer SVN repository is located under:

```
svn co https://svn.jboss.org/repos/tattletale/trunk/ tattletale-trunk
```

The project is compiled using Java Development Kit 1.5 or higher, Apache Ant 1.7 or higher and Apache Ivy 2.1 or higher. Using

```
ant <target>
```

where target is one of

- dist  
Builds the distribution.
- release  
Builds the release archives.
- doc  
Builds the documentation for the project.
- clean  
Cleans the project of temporary files.

See the full list of targets in the main build.xml file.

## 3.2. Configuration

The configuration of JBoss Tattletale is done through its

```
jboss-tattletale.properties
```

file.

The current configuration parameters includes:

**Table 3.1. JBoss Tattletale configuration**

Key	Value
reports	<p>A comma separated list of which reports that should be generated. The following reports are supported:</p> <ul style="list-style-type: none"><li>dependants The "Dependants" report.</li><li>dependson The "DependsOn" report.</li><li>graphviz The "Graphical dependencies" report.</li><li>transitivedependants The "Transitive Dependants" report.</li><li>transitivedependson The "Transitive DependsOn" report.</li><li>circulardependency The "Circular Dependency" report.</li><li>classlocation The "Class Location" report.</li><li>osgi The "OSGi" report.</li><li>sealed The "Sealed Information" report.</li><li>sign The "Signing Information" report.</li><li>eliminatejars</li></ul>

Key	Value
	<p>The "Eliminate Jar files with different versions" report.</p> <ul style="list-style-type: none"> <li><code>invalidversion</code></li> </ul> <p>The "Invalid version" report.</p> <ul style="list-style-type: none"> <li><code>multiplejars</code></li> </ul> <p>The "Multiple Jar files" report.</p> <ul style="list-style-type: none"> <li><code>multiplejarspackage</code></li> </ul> <p>The "Multiple Jar files (Package)" report.</p> <ul style="list-style-type: none"> <li><code>multiplelocations</code></li> </ul> <p>The "Multiple Locations" report.</p> <ul style="list-style-type: none"> <li><code>unusedjar</code></li> </ul> <p>The "Unused Jar" report.</p> <ul style="list-style-type: none"> <li><code>blacklisted</code></li> </ul> <p>The "Black listed" report.</p> <ul style="list-style-type: none"> <li><code>noversion</code></li> </ul> <p>The "No version" report.</p> <ul style="list-style-type: none"> <li><code>jar</code></li> </ul> <p>The "Jar archive" report.</p> <p>Default: All reports (<code>reports=*</code>)</p>
<code>classloader</code>	<p>Specifies which classloader structure that should be used when scanning the archives. Can be one of the following:</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure</code></li> </ul> <p>A no-operation classloader structure implementation that doesn't scope any archives.</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS4ClassLoaderStructure</code></li> </ul> <p>A classloader structure implementation that scopes based on JBoss Application Server 4.x directory structures.</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS5ClassLoaderStructure</code></li> </ul> <p>A classloader structure implementation that scopes based on JBoss Application Server</p>



Key	Value
	5.x directory structures.
profiles	<p>Specifies a comma separated list of profiles to resolve against. The following profiles are supported:</p> <ul style="list-style-type: none"> <li>java5 The Java 5 API.</li> <li>java6 The Java 6 API.</li> <li>ee5 The Java Enterprise Edition 5 API.</li> <li>ee6 The Java Enterprise Edition 6 API.</li> <li>seam22 The JBoss Seam 2.2 API.</li> <li>cdi10 The Contexts and Dependency Injection (CDI) 1.0 API.</li> <li>spring25 The Spring 2.5 API.</li> <li>spring30 The Spring 3.0 API.</li> </ul>
excludes	<p>A comma separated list of directories or files that should be excluded from the scan. F.ex.</p> <p><code>**/server/**,myjar.jar</code></p>
blacklisted	<p>A comma separated list of black listed classes or packages. F.ex.</p> <p><code>com.mycompany.forinternaluseonly, com.partner.forinternaluseonly</code></p>
scan	<p>A comma separated list of file extensions that should be scanned</p> <p>Default: <code>.jar</code></p>
graphvizDot	<p>The full path to the Graphviz DOT executable. This property is required if you want to generate PNG files and the Graphviz bin directory is not on your path. F.ex.</p>

Key	Value
	graphvizDot=C:\\Graphviz2.26.3\\bin\\dot.exe or graphvizDot=/opt/graphviz/bin/dot

The load order for the configuration file is

1. configuration parameter in the Apache Ant task
2. -Djboss-tattletale.properties system property
3. jboss-tattletale.properties file in current directory
4. jboss-tattletale.properties file in class loader

NOTE: The classloader structure feature is currently based on directory structures and may therefore fail to identify archives that should be included in the reports. If you want to be sure that all archives are included use the Noop-ClassLoaderStructure plugin.

### 3.2.1. Filtering

JBoss Tattletale supports filtering of the warnings and errors that the reports generates.

This functionality allows the user to filter out any warnings or errors that should be ignored and thereby allow the application to successful terminate.

The format of the filter properties file is

**Table 3.2. JBoss Tattletale filter**

Key	Value
report id	The filter

An example:

```
myreport=myfilter
```

The load order for the filter file is

1. filter parameter in the Apache Ant task
2. -Djboss-tattletale-filter.properties system property
3. jboss-tattletale-filter.properties file in current directory

See the individual reports for filtering support.

## 3.3. Running

Running JBoss Tattletale is very easy

```
java -Xmx512m -jar tattletale.jar [-exclude=<excludes>] <sourcedir> [<outputdir>]
```

where the "sourcedir" is the directory that contains your Java archives and the optional "outputdir" parameter is the directory where you would like your reports to be generated. The "-exclude" option let you exclude directories or file on the command line - see the configuration file for syntax.

The main file will be generated under the output directory as index.html.

JBoss Tattletale will scan for Java Archives (.JAR) files.

JBoss Tattletale requires Java Runtime Environment 5 or higher.

# 4

## Apache Ant

JBoss Tattletale integrates with Apache Ant such that you can generate the reports directly from your build environment.

First, you need to add tattletale.jar, tattletale-ant.jar and javassist.jar to the Apache Ant classpath.

Second, you need to add the following to your project definition tag:

```
xmlns:tattletale="antlib:org.jboss.tattletale.ant"
```

That is it.

Alternative, you can do a taskdef for each task

```
<taskdef name="report"
  classname="org.jboss.tattletale.ant.ReportTask"
  classpathref="tattletale.lib.path.id"/>
```

See the Apache Ant documentation for additional instructions on installation.

### 4.1. report

Usage:

```
<tattletale:report source="${src.dir}" destination="${dest.dir}"/>
```

**Table 4.1. Apache Ant: report**

Key	Value
source	The directory that contains the Java archives Default: Current directory
destination	The directory where the reports should be generated Default: Current directory

Key	Value
configuration	<p>Path to the configuration file</p> <p>Default: No value</p>
filter	<p>Path to the filter file</p> <p>Default: No value</p>
reports	<p>A comma separated list of which reports that should be generated. All reports can be selected by specifying "*". The following reports are supported:</p> <ul style="list-style-type: none"> <li>dependants The "Dependants" report.</li> <li>dependson The "DependsOn" report.</li> <li>graphviz The "Graphical dependencies" report.</li> <li>transitivedependants The "Transitive Dependants" report.</li> <li>transitivedependson The "Transitive DependsOn" report.</li> <li>circulardependency The "Circular Dependency" report.</li> <li>classlocation The "Class Location" report.</li> <li>osgi The "OSGi" report.</li> <li>sealed The "Sealed Information" report.</li> <li>sign The "Signing Information" report.</li> <li>eliminatejars</li> </ul>

Key	Value
	<p>The "Eliminate Jar files with different versions" report.</p> <ul style="list-style-type: none"> <li><code>invalidversion</code></li> </ul> <p>The "Invalid version" report.</p> <ul style="list-style-type: none"> <li><code>multiplejars</code></li> </ul> <p>The "Multiple Jar files" report.</p> <ul style="list-style-type: none"> <li><code>multiplejarspackage</code></li> </ul> <p>The "Multiple Jar files (Package)" report.</p> <ul style="list-style-type: none"> <li><code>multiplelocations</code></li> </ul> <p>The "Multiple Locations" report.</p> <ul style="list-style-type: none"> <li><code>unusedjar</code></li> </ul> <p>The "Unused Jar" report.</p> <ul style="list-style-type: none"> <li><code>blacklisted</code></li> </ul> <p>The "Black listed" report.</p> <ul style="list-style-type: none"> <li><code>noversion</code></li> </ul> <p>The "No version" report.</p> <ul style="list-style-type: none"> <li><code>jar</code></li> </ul> <p>The "Jar archive" report.</p> <p>Default: All reports</p>
<code>classloader</code>	<p>Specifies which classloader structure that should be used when scanning the archives. Can be one of the following:</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure</code></li> </ul> <p>A no-operation classloader structure implementation that doesn't scope any archives.</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS4ClassLoaderStructure</code></li> </ul> <p>A classloader structure implementation that scopes based on JBoss Application Server 4.x directory structures.</p> <ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS5ClassLoaderStructure</code></li> </ul> <p>A classloader structure implementation that scopes based on JBoss Application Server</p>

Key	Value
	<p>5.x directory structures.</p> <p>Default: <code>org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure</code></p>
profiles	<p>Specifies a comma separated list of profiles to resolve against. All profiles can be selected by specifying "*". The following profiles are supported:</p> <ul style="list-style-type: none"> <li>• <code>java5</code> The Java 5 API.</li> <li>• <code>java6</code> The Java 6 API.</li> <li>• <code>ee5</code> The Java Enterprise Edition 5 API.</li> <li>• <code>ee6</code> The Java Enterprise Edition 6 API.</li> <li>• <code>seam22</code> The JBoss Seam 2.2 API.</li> <li>• <code>cdi10</code> The Contexts and Dependency Injection (CDI) 1.0 API.</li> <li>• <code>spring25</code> The Spring 2.5 API.</li> <li>• <code>spring30</code> The Spring 3.0 API.</li> </ul> <p>Default: <code>java5, java6</code></p>
excludes	<p>A comma separated list of directories or files that should be excluded from the scan. F.ex.</p> <p><code>**/server/**,myjar.jar</code></p> <p>Default: Empty list</p>
blacklisted	<p>A comma separated list of black listed classes or packages. F.ex.</p> <p><code>com.mycompany.forinternaluseonly, com.partner.forinternaluseonly</code></p> <p>Default: Empty list</p>

Key	Value
failOnInfo	Fail the build if a failed INFO report is found  Default: <code>false</code>
failOnWarn	Fail the build if a failed WARN report is found  Default: <code>false</code>
failOnError	Fail the build if a failed ERROR report is found  Default: <code>false</code>
scan	A comma separated list of file extensions that should be scanned  Default: <code>.jar</code>

**Note**

Note that defining a property in the task overrides the setting in the configuration file.



# 5

## Apache Maven

JBoss Tattletale integrates with Apache Maven such that you can generate the reports directly from your build environment.

To be able to use the Tattletale Maven plugin in your Maven project, you will have to add the following plugin declaration in the pom.xml of your project:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jboss.tattletale</groupId>
      <artifactId>tattletale-maven</artifactId>
      <!-- The version of the plugin you want to use -->
      <version>1.1.0.Final</version>
      <executions>
        <execution>
          <goals>
            <goal>report</goal>
          </goals>
        </execution>
      </executions>
      <configuration>
        <!-- This is the location which will be scanned for generating tattletale reports -->
        <source>/absolutepath/to/source/dir</source>
        <!-- This is where the reports will be generated -->
        <destination>/absolute/path/to/reports/dir</destination>
      </configuration>
    </plugin>
  </plugins>
</build>
```

### Note

By default, the tattletale-maven plugin is attached to the "package" phase of Maven.

## 5.1. report

Usage:

Once you have configured your project's pom.xml to include the tattletale-maven plugin, as explained earlier, you can generate the report by running the package goal on your project

```
mvn clean package
```

**Table 5.1. Apache Maven: report**

Key	Value
source	The directory that contains the Java archives Default: Current directory
destination	The directory where the reports should be generated Default: Current directory
configuration	Path to the configuration file Default: No value
filter	Path to the filter file Default: No value
reports	<p>Contains nested <code>report</code> elements of which reports that should be generated. All reports can be selected by specifying <code>"*"</code>. The following reports are supported:</p> <ul style="list-style-type: none"> <li> <code>dependants</code>  The "Dependants" report. </li> <li> <code>dependson</code>  The "DependsOn" report. </li> <li> <code>graphviz</code>  The "Graphical dependencies" report. </li> <li> <code>transitivedependants</code>  The "Transitive Dependants" report. </li> <li> <code>transitivedependson</code>  The "Transitive DependsOn" report. </li> <li> <code>circulardependency</code>  The "Circular Dependency" report. </li> <li> <code>classlocation</code>  The "Class Location" report. </li> <li> <code>osgi</code>  The "OSGi" report. </li> </ul>

Key	Value
	<ul style="list-style-type: none"> <li>sealed The "Sealed Information" report.</li> <li>sign The "Signing Information" report.</li> <li>eliminatejars The "Eliminate Jar files with different versions" report.</li> <li>invalidversion The "Invalid version" report.</li> <li>multiplejars The "Multiple Jar files" report.</li> <li>multiplejarspackage The "Multiple Jar files (Package)" report.</li> <li>multiplelocations The "Multiple Locations" report.</li> <li>unusedjar The "Unused Jar" report.</li> <li>blacklisted The "Black listed" report.</li> <li>noverversion The "No version" report.</li> <li>jar The "Jar archive" report.</li> </ul> <p>Default: All reports</p>
classloader	<p>Specifies which classloader structure that should be used when scanning the archives. Can be one of the following:</p> <ul style="list-style-type: none"> <li>org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure A no-operation classloader structure implementation that doesn't scope any archives.</li> </ul>

Key	Value
	<ul style="list-style-type: none"> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS4ClassLoaderStructure</code> A classloader structure implementation that scopes based on JBoss Application Server 4.x directory structures.</li> <li><code>org.jboss.tattletale.reporting.classloader.JBossAS5ClassLoaderStructure</code> A classloader structure implementation that scopes based on JBoss Application Server 5.x directory structures.</li> </ul> <p>Default: <code>org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure</code></p>
<code>profiles</code>	<p>Contains nested <code>profile</code> elements of profiles to resolve against. All profiles can be selected by specifying <code>"*"</code>. The following profiles are supported:</p> <ul style="list-style-type: none"> <li><code>java5</code> The Java 5 API.</li> <li><code>java6</code> The Java 6 API.</li> <li><code>ee5</code> The Java Enterprise Edition 5 API.</li> <li><code>ee6</code> The Java Enterprise Edition 6 API.</li> <li><code>seam22</code> The JBoss Seam 2.2 API.</li> <li><code>cdi10</code> The Contexts and Dependency Injection (CDI) 1.0 API.</li> <li><code>spring25</code> The Spring 2.5 API.</li> <li><code>spring30</code> The Spring 3.0 API.</li> </ul> <p>Default: <code>&lt;report&gt;java5&lt;/report&gt;&lt;report&gt;java6&lt;/report&gt;</code></p>
<code>excludes</code>	<p>Contains nested <code>exclude</code> elements of directories or files that should be excluded from the scan. F.ex.</p>

Key	Value
	<code>&lt;exclude&gt;**/server/**&lt;/exclude&gt;&lt;exclude&gt;myjar.jar&lt;/exclude&gt;</code>  Default: Empty list
blacklisted	Contains nested <code>blacklist</code> elements of black listed classes or packages. F.ex. <code>&lt;blacklist&gt;com.mycompany.forinternaluseonly&lt;/blacklist&gt;</code> <code>&lt;blacklist&gt;com.partner.forinternaluseonly&lt;/blacklist&gt;</code>  Default: Empty list
failOnInfo	Fail the build if a failed INFO report is found  Default: <code>false</code>
failOnWarn	Fail the build if a failed WARN report is found  Default: <code>false</code>
failOnError	Fail the build if a failed ERROR report is found  Default: <code>false</code>
scan	A comma separated list of file extensions that should be scanned  Default: <code>.jar</code>

## Note

Note that defining a property in the task overrides the setting in the configuration file.

## 6.1. Dependency reports

### 6.1.1. Dependants

The dependants report will lists which archives depends on a specific archive.

**Table 6.1. Dependants report**

Archive	Dependants
The archive	A list of archives that depends on this archive

### 6.1.2. Depends On report

The depends on report will lists which archives that an archive depends on.

**Table 6.2. Depends On report**

Archive	Depends On
The archive	A list of archives which the archive depends on. Classes which can't be found are listed in italic

Filter key is: dependson

Filter definition is:

```
archive=[class|package](,[class|package])*;
```

An example:

```
dependson=myjar1.jar=org.eclipse.*;myjar2.jar=com.mycompany.MyClass,com.mycompany.OtherClass
```

### 6.1.3. Graphical dependencies report

The graphical dependencies report will create GraphViz dot files that show the dependencies as graphics.

As an example you can generate a PNG image using

```
dot -Tpng myarchive.dot > myarchive.png
```

See the GraphViz documentation for a full description on how to generate these images.

**Table 6.3. Graphical dependencies report**

Archive	Archives	Packages
The archive	GraphViz file that shows inter-archive dependencies	GraphViz file that shows inter-package dependencies

### 6.1.4. Transitive Dependants

The transitive dependants report will lists all archives depends on a specific archive.

**Table 6.4. Transitive Dependants report**

Archive	Dependants
The archive	A list of all archives that depends on this archive

### 6.1.5. Transitive Depends On report

The transitive depends on report will lists all archives that an archive depends on.

**Table 6.5. Transitive Depends On report**

Archive	Depends On
The archive	A list of all archives which the archive depends on. Classes which can't be found are listed in <i>italic</i>

Filter key is: `transitivedependson`

Filter definition is:

```
archive=[class|package](,[class|package])*;
```

An example:

```
transitivedependson=myjar.jar=com.mycompany.MyClass,com.mycompany.OtherClass
```

### 6.1.6. Circular Dependency report

The circular dependency report will lists all archives that has a circular dependency with another archive.

Archives that are marked with "(\*)" has the circular dependency. Note, that the circular dependency can be through a transitive dependency and not a direct dependency.

**Table 6.6. Circular Dependency report**

Archive	Circular Dependencies
The archive	A list of all archives which the archive has a circular dependency on.

Filter key is: `circulardependency`

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
circulardependency=myjar1.jar,myjar2.jar
```

## 6.2. General reports

### 6.2.1. Class Location

The class location will lists which archives that contain a specific class file.

**Table 6.7. Class Location report**

Class	Jar file
The class	The list of archives that contains the class

Filter key is: `classlocation`



Filter definition is:

```
[class|package](,[class|package])*;
```

An example:

```
classlocation=org.eclipse.*,com.mycompany.MyClass
```

## 6.2.2. OSGi

The OSGi report will display the OSGi state of your project.

**Table 6.8. OSGi report**

Archive	OSGi	Report	Manifest
The archive	The OSGi state of the archive	The OSGi report for the archive	A sample OSGi enabled MANIFEST file

Filter key is: osgi

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
osgi=myjar1.jar,myjar2.jar
```

## 6.2.3. Sealed information

The sealed information report will display the sealed status of your project.

**Table 6.9. Sealed information report**

Archive	Status
The archive	The status if the archive is sealed or unsealed

Filter key is: sealed

Filter definition is:

```
[yes|on|true|no|off|false]
```

An example:

```
sealed=off
```

## 6.2.4. Signing information

The signing information report will display the signing status of your project.

**Table 6.10. Signing information report**

Archive	Status
The archive	The status if the archive is signed or unsigned

Filter key is: `sign`

Filter definition is:

```
[yes|on|true|no|off|false]
```

An example:

```
sign=off
```

## 6.2.5. Eliminate Jar files with different versions

The eliminate jar files with different versions lists archives that have the same name but has a different version identifier.

**Table 6.11. Eliminate Jar report**

Archive	Location
The archive	The list of locations that the archive is found

Filter key is: `eliminatejars`

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
eliminatejars=myjar1.jar,myjar2.jar
```

## 6.2.6. Invalid version

The invalid version report lists archives that doesn't have a valid OSGi version identifier.

**Table 6.12. Invalid version report**

Name	Location
The archive name	The location and version identifier for the archive

Filter key is: `invalidversion`

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
invalidversion=myjar1.jar,myjar2.jar
```

## 6.2.7. Multiple Jar files

The multiple jar files report will list classes that appear in multiple jar files.

**Table 6.13. Multiple Jar files report**

Class	Jar files
The class	The list of archives where this class is found

Filter key is: `multiplejars`

Filter definition is:

```
[package](,[package])*;
```

An example:

```
multiplejars=com.mycompany.mypackage1,com.mycompany.mypackage2
```

## 6.2.8. Multiple Jar files (Package)

The multiple jar files fpr packages report will list packages that appear in multiple jar files.

**Table 6.14. Multiple Jar files report (Package)**

Package	Jar files
The package name	The list of archives where this package is found

Filter key is: `multiplejarspackage`

Filter definition is:

```
[package](,[package])*;
```

An example:

```
multiplejarspackage=com.mycompany.mypackage1,com.mycompany.mypackage2
```

## 6.2.9. Multiple Locations

The multiple locations report will list archives that appear in multiple locations under the scanned source directory.

**Table 6.15. Multiple Locations report**

Name	Location
The archive name	The list of locations where the archive is found

Filter key is: `multiplelocations`

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
multiplelocations=myjar1.jar,myjar2.jar
```

## 6.2.10. Unused Jar

The Unused Jar report lists archives that isn't referenced from any other Jar archive in the distribution. This doesn't mean however that the archive isn't used since it could be referenced through Java Reflection or through metadata.

**Table 6.16. Unused Jar report**

Archive	Used
The archive	Status if the archive is used or not

Filter key is: unusedjar

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
unusedjar=myjar1.jar,myjar2.jar
```

## 6.2.11. Black listed

The black listed report will list archives that uses black listed APIs.

**Table 6.17. Black listed report**

Archive	Usage
The archive name	The list of packages that uses black listed APIs

Filter key is: blacklisted

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
blacklisted=myjar1.jar,myjar2.jar
```

## 6.2.12. No version

The no version report will list archives that doesn't have a version identifier.

**Table 6.18. No version report**

Name	Location
The archive name	The list of locations where the archive is found

Filter key is: `noversion`

Filter definition is:

```
[archive](,[archive])*;
```

An example:

```
noversion=myjar1.jar,myjar2.jar
```

## 6.3. Archive reports

### 6.3.1. Java ARchive (JAR)

The Java ARchive (JAR) report will provide you with an overview of the archive.

**Table 6.19. No version report**

Key	Value
Name	The archive name
Class Version	The version identifier for the class files
Locations	The list of locations for the archive

Key	Value
Manifest	The manifest file
Signing information	The signing information for the archive
Requires	The list of required classes
Provides	The list of provided classes - including AssemblyVersion (if present)

## Troubleshooting

### 7.1. JBoss Tattletale generates empty reports

JBoss Tattletale generates its reports based on Java archives and not source code. Make sure that sourcedir you specify when running JBoss Tattletale contains the Java archives (f.ex. .JAR files) that you need scanned.

### 7.2. JBoss Tattletale throws an OutOfMemoryException

JBoss Tattletale needs to process the information it gathers in memory, so you need to provide enough memory for that to happen. You can adjust the -Xmx parameter of the command line below if you are using Sun's Java Runtime Environment.

```
java -Xmx1024m -jar jboss-tattletale.jar <sourcedir> [<outputdir>]
```

### 7.3. How do I ?

We can't cover every single issue in this guide, so feel free to drop by our forums to see if a solution has already been provided. Otherwise feel free to ask your question there.

Our forum is located at <http://www.jboss.org/index.html?module=bb&op=viewforum&f=320>  
[<http://www.jboss.org/index.html?module=bb&op=viewforum&f=306>]