JBoss Tattletale 1.0 User's Guide

Betraying all your project's naughty little secrets

Table of Contents

1

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.

Jay Balunas is a core developer on the JBoss Tattletale project. He can be reached at jay (dot) balunas (at) jboss (dot) org.

James Cobb does all the graphics for the JBoss Tattletale project. He can be reached at jcobb (at) redhat (dot) com.

2

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 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
- 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/.

3

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-1.0.0.GA.zip
```

or

```
tar xzf jboss-tattletale-1.0.0.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. 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 and Apache Ant 1.7 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
classloader	Specifies which classloader structure that should be used when scanning the archives. Can be one of the following: • org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure A no-operation classloader structure implementation that doesn't scope any archives. • org.jboss.tattletale.reporting.classloader.JBossAS4ClassLoaderStructure A classloader structure implementation that scopes based on JBoss Application

Key	Value
	Server 4.x directory structures. • org.jboss.tattletale.reporting.classloader.JBossAS5ClassLoaderStructure A classloader structure implementation that scopes based on JBoss Application Server 5.x directory structures.
outputDir	The output directory where the report gets generated.
blacklisted	A comma separated list of black listed classes or packages. F.ex. com.mycompany.forinternaluseonly, com.partner.forinternaluseonly

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.

3.3. Running

Running JBoss Tattletale is very easy

```
java -Xmx512m -jar jboss-tattletale.jar <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 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.

3.4. Apache Ant integration

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

First, you need to add jboss-tattletale.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. See the Apache Ant documentation for additional instructions on installation.

3.4.1. report

Usage:

```
<tattletale:report scanDir="" outputDir=""/>
```

where scanDir is the directory that contains the Java archives and outputDir is the directory where the reports should be generated.

4

Reports

4.1. Dependency reports

4.1.1. Dependants

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

Table 4.1. Dependants report

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

4.1.2. Depends On report

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

Table 4.2. 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

4.1.3. 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 4.3. Depends On report

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

4.1.4. Transitive Dependants

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

Table 4.4. Transitive Dependants report

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

4.1.5. Transitive Depends On report

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

Table 4.5. 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 italic

4.2. General reports

4.2.1. Class Location

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

Table 4.6. Class Location report

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

4.2.2. OSGi

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

Table 4.7. 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

4.2.3. 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 4.8. Eliminate Jar report

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

4.2.4. Invalid version

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

Table 4.9. Invalid version report

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

4.2.5. Multiple Jar files

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

Table 4.10. Multiple Jar files report

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

4.2.6. Multiple Locations

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

Table 4.11. Multiple Locations report

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

4.2.7. Black listed

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

Table 4.12. Black listed report

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

4.2.8. No version

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

Table 4.13. No version report

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

4.3. Archive reports

4.3.1. Java ARchive (JAR)

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

Table 4.14. No version report

Key	Value
Name	The archive name
Locations	The list of locations for the archive
Manifest	The manifest file
Requires	The list of required classes
Provides	The list of provided classes - including SerialVersionUID (if present)

5

Troubleshooting

5.1. 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.

5.2. 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 ajust 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>]
```

5.3. 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=306