JBoss Tattletale 1.2 Developer's Guide

Betraying all your project's naughty little secrets

Copyright © 2011 Red Hat Middleware

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.

1.2. Thanks to

Jay Balunas, James Cobb, Torben Jaeger, Navin Surtani, Rostislav Svoboda and Steve Taranto.

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/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
- And generate the same reports for your .WAR and .EAR archives

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

Building

3.1. Prerequisites

3.1.1. Java Development Kit (JDK)

You must have one of the following JDKs installed in order to build the project:

- Sun JDK 1.5.x
- Sun JDK 1.6.x

Remember to ensure that "javac" and "java" are in your path (or symlinked).

```
JAVA_HOME=/location/to/javahome
export JAVA_HOME

PATH=$JAVA_HOME/bin:$PATH
export PATH
```

3.1.2. Apache Ant

You must have Apache Ant 1.8.1+ installed on your system.

Remember to ensure that "ant" are in your path (or symlinked).

```
ANT_HOME=/location/to/anthome
export ANT_HOME

PATH=$ANT_HOME/bin:$PATH
export PATH
```

3.1.3. Apache Ivy

JBoss Tattletale uses Apache Ivy for dependency management. The build environment automatically bootstrap the installation of Apache Ivy.

3.1.4. GitHub

You must have git installed on your system.

Remember to ensure that "git" are in your path (or symlinked).

3.2. Obtaining the source code

3.2.1. GitHub

You can clone the repository onto your system by the following command. This automatically creates a directory 'tattletale' within your current directory. Please note that this would be a read-only directory.

```
git clone git://github.com/jesperpedersen/tattletale.git
```

Should you want to edit the sources yourself, it is best to create your own fork of the repository and then clone that particular fork on your command line. Should you need help with Git and GitHub then read the help files (http://help.github.com/)

3.3. Compiling the source code

In order to build the JBoss Tattletale project you execute:

```
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.

4

Coding

4.1. Source code formatting

JBoss Tattletale uses 3 spaces for indentation with curly brakets on the next line as the class or the statement.

```
/**
 * My class
 */
public class MyClass
{
  /**
   * My method
   */
  public void myMethod()
  {
    if (value)
    {
      // Do something
    }
    else
    {
      // Do something else
    }
  }
}
```

Remember to add JavaDoc for all classes and methods.

You can verify the formatting rules by running checkstyle on the source code:

```
ant checkstyle
```

See the source code for more examples of the source code formatting rules.

4.2. Adding a new report

Adding a new report to JBoss Tattletale is very easy, so if you have found a report that you would like to see that shouldn't stop you from adding it.

You need a class that implements:

```
org.jboss.tattletale.reporting.Report
```

The JavaDoc for JBoss Tattletale will show you the class hierarchy as you may want to extend from another class (org.jboss.tattletale.reporting.AbstractReport) for example.

After you have implemented the new report you add it to the properties file 'jboss-tattletale.properties'. For example:

```
   customreport.1=my.custom.package.customreport
```

At runtime, as long as your custom jar file is in the classpath your custom report will be generated. If you were running from the command line for example:

```
  java -classpath tattletale.jar:customreporting.jar org.jboss.tattletale.Main [directory-of-source-jar] [output-directory]
```

4.3. Adding a new classloader structure

JBoss Tattletale include functionality to identify certain classloader structures.

The plugins must implement the

```
 org.jboss.tattletale.reporting.classloader.ClassLoaderStructure
```

interface and contain a default no-argument constructor.

The interface only contain one method that needs to be implemented:

```
/**
 * Can one archive see the other
 * @param from The from archive
 * @param to The to archive
 * @return True if from can see to; otherwise false
 */
public boolean isVisible(Archive from, Archive to);
```

which returns true if the "from" archive can see the "to" archive. Otherwise false should be returned.

The current plugins can be used as a starting point for developing a new plugin:

- org.jboss.tattletale.reporting.classloader.NoopClassLoaderStructure

  A no operation plugin that always will include the queried archive in the report.

- org.jboss.tattletale.reporting.classloader.JBossAS4ClassLoaderStructure

  Plugin for the JBoss Application Server 4.x series.

- org.jboss.tattletale.reporting.classloader.JBossAS5ClassLoaderStructure

  Plugin for the JBoss Application Server 5.x series.

The plugin is loaded through the 'classloader' key in jboss-tattletale.properties file. See the user guide for more information.