

JBoss Profiler 2.0 User's Guide

Copyright © 2009 Red Hat Middleware

Table of Contents

1. About JBoss Profiler 2	1
1.1. The team	1
1.1.1. Contributors	1
2. Introduction	2
3. Installation	4
3.1. JBoss Application Server 5.x and 6.x	4
3.2. JBoss Application Server 4.2	4
3.3. Source code	5
4. Configuration	7
5. Command line client	11
5.1. Overview	11
5.2. Commands	11
5.2.1. startProfiler	11
5.2.2. stopProfiler	11
5.2.3. snapshot	12
5.2.4. getSnapshot	12
5.2.5. listSnapshots	12
5.2.6. clearSnapshots	12
5.2.7. gc	13
5.2.8. enable	13
5.2.9. disable	13
5.2.10. load	13
5.2.11. save	13
5.2.12. diff	14
5.2.13. add	14
5.2.14. remove	14
5.2.15. list	14
5.3. Configuration	14
5.4. Running the example	15
5.5. Using the precompiler	16
6. Apache Ant integration	17
6.1. Profiler tasks	17
6.2. Regression tasks	18
6.2.1. Regression task	18
7. JBoss Application Server	20
7.1. Communicator MBean	20
7.2. Twiddle	20
7.2.1. Information	20
7.2.2. Operations	21
8. Reports	22
8.1. Overview report	22
8.2. Packages report	23
8.3. Classes report	23
8.4. Methods report	23

8.5. Hotspots report	23
8.6. Caller report	24
8.7. Wait time report	24
8.8. PerThread report	24
8.9. PerClass report	24
9. Troubleshooting	26
9.1. How do I ?	26

About JBoss Profiler 2

JBoss Profiler 2 is a 100% pure Java 5 profiler capable of profiling Java Standard Edition and Java Enterprise Edition applications.

In development are currently a web front-end using JSF/RichFaces and a JVMTI native backend.

1.1. The team

Jesper Pedersen acts as the lead for the JBoss Profiler 2 project. He can be reached at [jesper \(dot\) pedersen \(at\) jboss \(dot\) org](mailto:jesper@jboss.org).

1.1.1. Contributors

Huijuan Shao is a major contributor for the JSF/RichFaces web front-end. He can be reached at [hjshao \(at\) hitachi \(dot\) cn](mailto:hjshao@hitachi.cn).

2

Introduction

JBoss Profiler 2 is a 100% pure Java profiler capable of profiling Java Enterprise 5 applications.

- 100% Pure Java
- Integration with JBoss Enterprise Middleware Suite (JEMS)
- Access through multiple protocols using JBoss Remoting
 - Socket
 - Remote Method Invocation (RMI)
 - Hyper Text Transport Protocol (HTTP)
- Reports
 - Overview
 - Packages
 - Classes
 - Methods
 - Hotspots
 - Caller
 - Wait time
 - PerThread
 - PerClass
- Able to specify method visibility
 - Global
 - Package
 - Method
- Component identification
 - Plain Old Java Object (POJO)
 - java.lang.Throwable
 - Enterprise JavaBean: Entity
 - Enterprise JavaBean: Session
 - Enterprise JavaBean: Message
 - Servlet
 - Servlet filter
 - JavaServer Pages
 - JMX MBean
 - JavaServer Faces Converter
 - JavaServer Faces Managed Bean
 - RMI: Remote
 - RMI: Server
 - CORBA: Object

- CORBA: Servant
- 3rd party plugins
- Compare snapshots
- Client
 - Apache Ant integration
 - Command line

JBoss Profiler 2 was designed to run on the JBoss Application Server 4.2, JBoss Application Server 5.x and JBoss Application Server 6.x releases or any other Java applications using Java Runtime Environment 5 or higher.

3

Installation

JBoss Profiler 2 can be downloaded in its binary form for easy installation.

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

Once downloaded extract the files by executing:

```
unzip jboss-profiler-2.0.0.GA.zip
```

or

```
tar xzf jboss-profiler-2.0.0.GA.tar.gz
```

depending on which archive type you downloaded.

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

3.1. JBoss Application Server 5.x and 6.x

- Copy jboss-profiler.jar to jbossas/bin
- Copy jboss-profiler.properties to jbossas/bin
- Edit jboss-profiler.properties in jbossas/bin to include the classes to be profiled
- Copy jboss-profiler-plugins.jar to jbossas/bin
- Edit run.conf (Unix) or run.bat (Windows) in jbossas/bin to include JBoss Profiler in JAVA_OPTS
- Copy jboss-profiler.sar to jbossas/server/<conf>/deploy
- Boot application server

Include the following in JAVA_OPTS

```
-javaagent:jboss-profiler.jar -Djboss-profiler.properties=jboss-profiler.properties
```

3.2. JBoss Application Server 4.2

- Copy jboss-profiler.jar to jbossas/bin
- Copy jboss-profiler.properties to jbossas/bin
- Copy javassist.jar to jbossas/bin

- Edit jboss-profiler.properties in jbossas/bin to include the classes to be profiled
- Copy jboss-profiler-plugins.jar to jbossas/bin
- Edit run.conf (Unix) or run.bat (Windows) in jbossas/bin to include JBoss Profiler in JAVA_OPTS
- Copy jboss-profiler.sar to jbossas/server/<conf>/deploy
- Boot application server

Include the following in JAVA_OPTS

```
-javaagent:jboss-profiler.jar -Djboss-profiler.properties=jboss-profiler.properties
```

3.3. 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/jbossprofiler/branches/JBossProfiler2 jboss-profiler-2
```

The developer SVN repository is located under:

```
svn co https://svn.jboss.org/repos/jbossprofiler/branches/JBossProfiler2 jboss-profiler-2
```

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.
- precompiler
Builds the precompiler.

- web

Builds the web front-end for the project.

- clean

Cleans the project of temporary files.

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

4

Configuration

The configuration of JBoss Profiler is done through its

```
jboss-profiler.properties
```

file.

Table 4.1. JBoss Profiler configuration

Key	Value
enable	Enable / disable profiler. [yes no] default: yes.
precompiled	Use precompiled archives. [yes no] default: no.
cpu	Enable / disable CPU profiling. [yes no] default: yes.
memory	Enable / disable memory profiling. [yes no] default: yes.
includes	Classes that should be profiled. [package class][[visibility]][,...] f.ex: org.jboss.profiler.test.* public,org.jboss.profiler.test.sub.* private
excludes	Classes that should excluded. [package class][,...] f.ex: *

Key	Value
visibility	<p>The method visibility.</p> <p>[public package protected private] default: private.</p>
save	<p>Save snapshots on exit.</p> <p>[yes no] default: yes.</p>
savelocation	<p>Location where to save the snapshots.</p> <p>[Path] default: current working directory.</p>
startup	<p>Start profiler during startup.</p> <p>[yes no] default: yes.</p>
repository	<p>Repository for class files. Must be enabled in order to add/remove classes at run-time.</p> <p>[yes no] default: no.</p>
remote	<p>Enable / disable remote API - this information is used when profiling Java SE applications.</p> <p>[yes no] default: yes.</p>
store	<p>Where should the profile information be stored in the agent.</p> <p>[memory file] default: memory.</p>
location	<p>The location for the file store. If not specified will default to java.io.tmpdir.</p> <p>Default: current directory.</p> <p>f.ex: /tmp/profiles</p>
host	<p>Name of binding address - this information is used when profiling Java SE applications.</p> <p>f.ex: localhost</p>
port	<p>Port number (socket) - this information is used when profiling Java SE applications.</p> <p>f.ex: 5400</p>

Key	Value
ejb	Identify Enterprise JavaBeans components. [yes no] default: yes.
servlet	Identify Servlet components. [yes no] default: yes.
jsf	Identify JavaServer Faces components. [yes no] default: yes.
jmx	Identify JMX MBean. [yes no] default: yes.
rmi	Identify RMI components. [yes no] default: yes.
corba	Identify CORBA components. [yes no] default: yes.
plugin.<number>	Plugin to load. Class file. f.ex: plugin.1=org.jboss.profiler.plugins.Hibernate
job.<number>.type	The job type. Class file. f.ex: job.1.type=org.jboss.profiler.agent.jobs.StartProfiling <ul style="list-style-type: none"> Start profiling: org.jboss.profiler.agent.jobs.StartProfiling Stop profiling: org.jboss.profiler.agent.jobs.StopProfiling
job.<number>.time	The job time in millis. The specified time represents a date. long. f.ex: job.1.time=1167609600000
job.<number>.period	The job execution period in millis. The jobs are executed at fixed rate.

Key	Value
	<p>If not specified the job is only executed once..</p> <p>long.</p> <p>f.ex: job.1.period=3600000</p>

Command line client

The suite features a command line client where you are able to control the operation of the agent as well as generating the profiling reports.

5.1. Overview

```
Usage: Client [-h host] [-p port] <command>
  startProfiler  : Start the profiler
  stopProfiler  : Stop the profiler
  snapshot      : Take a snapshot
  getSnapshot   : Get a snapshot
  listSnapshots : List snapshots
  clearSnapshots : Clear snapshots
  gc            : Trigger garbage collection
  enable        : Enable the profiler
  disable       : Disable the profiler
  load          : Load a snapshot
  save          : Save a snapshot
  diff          : Compare snapshots
  add           : Add classes (public|package|protected|private)
  remove        : Remove classes
  list          : List classes

Host defaults to 'localhost'.
Port defaults to '5400'.
```

The default extension for the generated snapshot is JBoss Profiler Snapshot (.JPS).

5.2. Commands

5.2.1. startProfiler

Start the profiler.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \
-jar jboss-profiler-client.jar startProfiler
```

5.2.2. stopProfiler

Stop the profiler.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar stopProfiler
```

5.2.3. snapshot

Take a snapshot. An optional argument will generate the reports in the specified directory.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar snapshot
```

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar snapshot reports
```

5.2.4. getSnapshot

Get a snapshot. An optional second argument will generate the reports in the specified directory.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar getSnapshot 1
```

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar getSnapshot 1 reports
```

5.2.5. listSnapshots

List available snapshots.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar listSnapshots
```

5.2.6. clearSnapshots

Clear all snapshots.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar clearSnapshots
```

5.2.7. gc

Trigger garbage collection.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar gc
```

5.2.8. enable

Enable the profiler.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar enable
```

5.2.9. disable

Disable the profiler.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar disable
```

5.2.10. load

Load a snapshot from either a file or a directory. This command will generate all the reports. An optional second argument will generate the reports in the specified directory.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar load 20070806144635343-20070806144638444.jps
```

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar load 20070806144635343/
```

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar load 20070806144635343-20070806144638444.jps reports
```

5.2.11. save

Save a snapshot to its default name or the specified name.


```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar save 1
```

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar save 1 /path/to/mysnapshot.jps
```

5.2.12. diff

Compare two snapshots.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar diff oldsnapshot.jps newsnapshot.jps
```

5.2.13. add

Add classes that should be profiled.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar add org.jboss.profiler.test.* public
```

5.2.14. remove

Remove classes that should not be profiled anymore.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar remove org.jboss.profiler.test.*
```

5.2.15. list

List classes being profiled.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar list
```

5.3. Configuration

The configuration of the command client is done in

```
jboss-profiler-client.properties
```

Table 5.1. JBoss Profiler command line client configuration

Key	Value
protocol	Protocol to use. [socket rmi http] default: socket.
host	The host name. Default: localhost.
port	The port number. Default: 5400.
threshold	Method threshold in ms. Default: 1.0.
savelocation	Location where to save the snapshots. Path Default: current working directory.
plugin.<number>	Plugin to load. Class plugin.1=org.jboss.profiler.plugins.Hibernate

5.4. Running the example

The example in the SVN repository can be run with:

```
java -javaagent:jboss-profiler.jar=remote=no \
-Djboss-profiler.properties=jboss-profiler.properties \
-classpath jboss-profiler.jar:jboss-profiler-connectors.jar:jboss-profiler-test.jar:\
          jbossall-client.jar:servlet.jar:ejb3-persistence.jar:jboss-ejb3x.jar:. \
  org.jboss.profiler.test.Test
```

5.5. Using the precompiler

Warning

The JBoss Profiler Precompiler will overwrite the JAR files in your application, so make sure that you keep a copy of the original files.

The JBoss Profiler Precompiler will precompile the JAR archives in the specified directory with instrumentation information used by the agent. This allows you to start up your application quicker.

The precompiler is run using the following command:

```
java -Xmx1024m -jar jboss-profiler-precompiler.jar \  
  <includes> <excludes> <visibility> <file|directory>
```

The includes, excludes and visibility flags are in the same format as the flags for the agent. An example:

```
java -Xmx1024m -jar jboss-profiler-precompiler.jar \  
  com.mycompany.* com.mycompany.highperformance.*,* private ~/myapplication
```

The precompiler will output any classes it can't precompile due to missing dependencies and the statistics for the run.

Apache Ant integration

6.1. Profiler tasks

The jboss-profiler-ant.jar file contains Ant tasks for the JBoss Profiler.

Assuming that this file has been placed in in \${ant.home}/lib the following file will automatically load the definitions of the tasks:

```
<project default="defaulttarget" xmlns:jboss-profiler="antlib:org.jboss.profiler.ant">
```

Otherwise you can configure the Ant integration using the following task definitions:

```
<taskdef name="enable" classname="org.jboss.profiler.ant.EnableProfilerTask"/>
<taskdef name="disable" classname="org.jboss.profiler.ant.DisableProfilerTask"/>
<taskdef name="startprofiler" classname="org.jboss.profiler.ant.StartProfilerTask"/>
<taskdef name="stopprofiler" classname="org.jboss.profiler.ant.StopProfilerTask"/>
<taskdef name="clearsnapshots" classname="org.jboss.profiler.ant.ClearSnapshotsTask"/>
<taskdef name="gc" classname="org.jboss.profiler.ant.GCTask"/>
<taskdef name="save" classname="org.jboss.profiler.ant.SaveTask"/>
<taskdef name="add" classname="org.jboss.profiler.ant.AddClassesTask"/>
<taskdef name="remove" classname="org.jboss.profiler.ant.RemoveClassesTask"/>
```

All tasks have the following attributes in common:

Table 6.1. Common Apache Ant elements

Element	Description
protocol	The protocol used for connection (socket, rmi, http) (default: socket).
host	The host name (default: localhost).
port	The port number (default: 5400)

The save task has these additional elements

Table 6.2. Save task

Element	Description
snapshot	The snapshot number.
destination	The destination of the snapshot.

The add task has these additional elements

Table 6.3. Add task

Element	Description
classes	The classes that should be added. [package class][[visibility]][,...] f.ex: org.jboss.profiler.test.* public,org.jboss.profiler.test.sub.* private

The remove task has these additional elements

Table 6.4. Remove task

Element	Description
classes	The classes that should be removed. [package class][,...] f.ex: org.jboss.profiler.test.*

Example:

```
<startprofiler protocol="socket"
               host="localhost"
               port="5400"/>
```

6.2. Regression tasks

The jboss-profiler-ant.jar file contains the following regression Ant tasks.

```
<taskdef name="regression" classname="org.jboss.profiler.ant.RegressionTask"/>
```

6.2.1. Regression task

Table 6.5. Regression task

Element	Description
oldsnapshot	The location of the old snapshot.
newsnapshot	The location of the new snapshot.
difference	The maximum difference in percent between the two snapshots.

Example:

```
<regression oldsnapshot="old.jps"
  newsnapshot="new.jps"
  difference="3"/>
```

JBoss Application Server

7.1. Communicator MBean

Configuration can be found in

```
jboss-profiler.sar/META-INF/jboss-service.xml
```

Table 7.1. Communicator configuration

Key	Description
Socket	Should socket communication be enabled ? [true false] default: true
SocketPort	The socket port default: 5400
Rmi	Should RMI communication be enabled ? [true false] default: false
RmiPort	The RMI port default: 5401
Http	Should HTTP communication be enabled ? [true false] default: false
HttpPort	The HTTP port default: 5402

7.2. Twiddle

7.2.1. Information

```
./twiddle.sh get 'org.jboss.profiler:service=Profiler' Running
./twiddle.sh get 'org.jboss.profiler:service=Profiler' Enabled
./twiddle.sh get 'org.jboss.profiler:service=Profiler' Memory
./twiddle.sh get 'org.jboss.profiler:service=Profiler' File
./twiddle.sh get 'org.jboss.profiler:service=Profiler' EJB
./twiddle.sh get 'org.jboss.profiler:service=Profiler' Servlet
./twiddle.sh get 'org.jboss.profiler:service=Profiler' JSF
./twiddle.sh get 'org.jboss.profiler:service=Profiler' JMX
./twiddle.sh get 'org.jboss.profiler:service=Profiler' RMI
./twiddle.sh get 'org.jboss.profiler:service=Profiler' CORBA
```

7.2.2. Operations

```
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' startProfiler
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' stopProfiler
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' enableProfiler
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' disableProfiler
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' clearSnapshots
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' addClasses org.jboss.test.* public
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' removeClasses org.jboss.test.*
./twiddle.sh invoke 'org.jboss.profiler:service=Profiler' listClasses
```


8

Reports

The load command generates the reports that will provide you with the information about the application that you profiled.

```
java -Xmx512m -Djboss-profiler-client.properties=jboss-profiler-client.properties \  
-jar jboss-profiler-client.jar load 20070806144635343-20070806144638444.jps
```

The main report will be located in the generated directory as index.html.

The generated reports uses to following abbreviations for the headlines

Table 8.1. Common abbreviations

Key	Description
Count	The number of invocations
Ms	The number of miliseconds
Ms (T)	The total number of miliseconds
Ms (N)	The number of miliseconds for this method
Avg	The average execution time in miliseconds
%	The percent
% (T)	The total percent
% (C)	The percent of the class
% (N)	The percent for this method
Method	The method name

8.1. Overview report

Provides an overview of all reports.

Information:

- Snapshot

- Thread
- Most Time (Top 10)
- Hotspots (Top 10)
- Allocation
- Class

8.2. Packages report

List package information.

Information:

- %
- Ms
- Package

8.3. Classes report

List class information.

Information:

- Class
- Type
- Time

8.4. Methods report

List method information.

Information:

- Count
- Ms
- Avg
- %
- Method

8.5. Hotspots report

List hotspots.

Information:

- Count
- Ms
- Avg

- %
- Method

8.6. Caller report

List callers - the report is linked from the PerClass report.

Information:

- Count
- Avg
- Ms
- Method

8.7. Wait time report

List the wait time.

Information:

- Class
- Type
- Time

8.8. PerThread report

List the call strack for a thread - the report is linked from the Overview report.

Information:

- Thread
- Count
- Ms (T)
- Ms (N)
- % (T)
- % (N)
- Method

8.9. PerClass report

List an overview of a class.

Method information:

- Count
- Ms
- % (C)

- % (T)
- Method

Hotspot information:

- Count
- Ms
- Avg
- %
- Method

9

Troubleshooting

9.1. 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=199>