



# JOnAS 4.8 EE Commands Reference Guide

*This document describes all the commands that can be used with JOnAS*

JOnAS Team ()

- 2007 -

Copyright © ObjectWeb 2007

---

## Table of Contents

I. jonas command .....	3
jonas start .....	4
jonas stop .....	5
jonas admin .....	6
jonas check .....	8
II. jclient command .....	11
jclient .....	12
III. GenIC command .....	13
GenIC .....	14
IV. JMS commands .....	16
JmsServer .....	17
joram_raconfig .....	18
V. RAConfig command .....	19
RAConfig .....	20
VI. newjb command .....	22
newjb .....	23

---

# jonas command

This command provides the capability to start, stop, or administrate JOnAS servers.

The following two scripts can be reviewed and possibly modified for assistance with problems or for obtaining additional information:

- **jonas** for UNIX systems
- **jonas.bat** for WINDOWS systems

There are five different sub-commands, that depend on the first mandatory argument: start, stop, admin, check, version

---

## Name

`jonas start` — comand allowing to start a JOnAS server

## Synopsis

```
jonas start [-fg | -bg | -win] [-n name] [-Ddomain.name=domain] [-target server]
```

## Description

Start a new JOnAS server.

The process can be run in the foreground, in the background, or in a new window. If the background option is chosen (default option), control is given back to the caller only when the server is ready.

The server's name is by default *jonas*. A different name can be given with the `-n` option.

The name of the management domain to which the server belongs is given by the *domain.name* property, or by the server's name if this property is not defined.

The name of the server has to be unique in the domain.

The `-target` option allows to start another server or cluster (group of servers) in the domain.

This action is to be executed in the environment of a running master server who's name is given by the `-n` option. In order to start a target server or cluster, the following conditions have to be met:

- the target must belong to the domain (has to be defined in the domain's map)
- a cluster daemon has to be running on the target's host and it has to be aware of the target (see cluster daemon configuration)

---

## Name

`jonas stop` — comand allowing to stop a JOnAS server

## Synopsis

```
jonas stop [-n name] [-target server]
```

## Description

Stop a running JOnAS server. Use the `-n` option if the server to stop was given a name other than the default name.

The server's name is by default *jonas*. A different name can be given with the `-n` option.

The name of the management domain to which the server belongs is given by the *domain.name* property, or by the server's name if this property is not defined.

The name of the server has to be unique in the domain.

The `-target` option allows to start another server or cluster (group of servers) in the domain. This action is to be executed in the environment of a running master server who's name is given by the `-n` option. In order to start a target server or cluster, the following conditions have to be met:

- the target must belong to the domain (has to be defined in the domain's map)
- a cluster daemon has to be running on the target's host and it has to be aware of the target (see cluster daemon configuration)

---

## Name

jonas admin — command allowing to administer a JOnAS server

## Synopsis

```
jonas admin [-n name] [admin_options] [-target server]
```

## Description

Administrate a JOnAS server. Use the `-n` option if the server was given a name other than the default name. Used without any other option, this command will prompt the user for an administrative command (interactive mode). Each administrative command exists in a non-interactive mode, for use in shell scripts or bat scripts, for example.

When using the `-target` option, the administrative commands are applied to another server or to a cluster (group of servers) running in the domain. This supposes that the command is executed in the master server's environment. The master server's name is specified with the `-n` option.

## Admin options

[? ]: Prints a help with all possible options.

[-a filename ]: Deploys a new application described by *filename* on the current JOnAS server, or on another target in the domain, if the current server is a master. The current server is identified by the default name or by the `-n` option.

The application can be one of the following:

- a standard **ejb-jar** file. This will lead to the creation of a new EJB container in the JOnAS server. If the file name has a relative path, this path is relative to where the server has been launched or relative to the `$JONAS_BASE/ejbjars` directory.
- a standard **.war** file containing a WEB module. If the file name has a relative path, this path is relative to where the server has been launched or relative to the `$JONAS_BASE/webapps` directory.
- a standard **.ear** file containing a complete J2EE application. If the file name has a relative path, this path is relative to where the server has been launched or relative to the `$JONAS_BASE/apps` directory.
- a standard **.rar** file containing a RAR module. If the file name has a relative path, this path is relative to where the server has been launched or relative to the `$JONAS_BASE/rars` directory.

**-r filename** : Dynamically undeploys a previously deployed application from the current server or from the specified target if the current server is a master.

**-gc** : Runs the garbage collector in the current JOnAS server.

**-passivate** : Passivates all entity bean instances. This affects only instances outside transaction.

**-e** : Lists the properties of the current JOnAS server.

**-j** : Lists the registered JNDI names, as seen by the current JOnAS server.

**-l** : Lists the beans currently loaded by the current JOnAS server.

**-sync** : Synchronizes the entity bean instances on the current JOnAS server. Note that this affects only the instances that are not involved in a transaction.

**-debug topic** : Sets the topic level to DEBUG.

**-tt timeout** : Changes the default timeout for transactions. timeout is in seconds.

**-ping [-timeout seconds]** : Waits until the JOnAS server is available.

Each jonas admin option has its equivalent in the interactive mode. To enter interactive mode and access the following list of subcommands, type `jonas admin [-n name]` without any other argument. To exit from interactive mode, use the exit command.

<b>interactive command</b>	<b>options</b>
<b>addbeans</b>	<b>-a filename</b>
<b>env</b>	<b>-e</b>
<b>gc</b>	<b>-gc</b>
<b>help</b>	<b>-?</b>
<b>jndinames</b>	<b>-j</b>
<b>listbeans</b>	<b>-l</b>
<b>removebeans</b>	<b>-r filename</b>
<b>sync</b>	<b>-sync</b>
<b>trace</b>	<b>-debug topic</b>
<b>timeout</b>	<b>-tt timeout</b>
<b>quit</b>	exit interactive mode

---

## Name

jonas check — command allowing to check JOnAS is well installed

## Synopsis

```
jonas check
```

## Description

Check JOnAS environment is correctly set.

If everything is set correctly, this command ends by displaying something that looks like the following output:

```
JONAS_BASE set to /home/coqp/jb
- JONAS_ROOT value:
/home/coqp/JONAS_4_8_5
- JONAS_BASE value:
/home/coqp/JONAS_4_8_5
- JOnAS Services:
registry,jmx,jtm,db,dbm,security,resource,ejb,ws,web,ear
- Contents of 'jonas.properties':
jonas.service.security.realm.jndi.registration = false
jonas.service.db.user1 = jonas:jonas
jonas.service.mail.factories =
jonas.service.jtm.remote = false
jonas.service.jms.mom = org.objectweb.jonas_jms.JmsAdminForJoram
jonas.service.ejb.auto-genic = true
jonas.service.resource.threadwaittimeout = 60
jonas.service.ejb.descriptors =
jonas.service.mail.class = org.objectweb.jonas.mail.MailServiceImpl
jonas.service.resource.resources =
jonas.service.web.descriptors =
jonas.service.ws.factory.class = org.objectweb.jonas.ws.axis.JAxisServiceFactory
jonas.security.manager = true
jonas.service.ws.wsdlhandlers = file1
jonas.service.ha.jgroups.conf = jgroups-ha.xml
jonas.service.registry.mode = collocated
jonas.service.dbm.datasources = HSQL1
jonas.service.resource.class = org.objectweb.jonas.resource.ResourceServiceImpl
jonas.service.security.class = org.objectweb.jonas.security.JonasSecurityServiceImpl
jonas.service.resource.parsingwithvalidation = true
jonas.service.discovery.class = org.objectweb.jonas.discovery.DiscoveryServiceImpl
jonas.service.ear.descriptors =
jonas.service.thread.class = org.objectweb.area.jonas.AreaService
jonas.service.resource.execworktimeout = 0
jonas.service.registry.class = org.objectweb.jonas.registry.RegistryServiceImpl
jonas.service.jms.queues = sampleQueue
jonas.service.discovery.multicast.address = 224.224.224.224
jonas.service.db.port = 9001
jonas.service.ejb.parsingwithvalidation = true
jonas.service.web.parsingwithvalidation = true
jonas.service.security.context.check.keystoreFile = /tmp/keystore
jonas.service.thread.ejbareaname = EJB
jonas.service.db.dbname = db_jonas
jonas.service.ejb.autoload = autoloader
jonas.service.jms.class = org.objectweb.jonas.jms.JmsServiceImpl
jonas.service.security.propagation = true
jonas.service.discovery.multicast.port = 9080
jonas.service.security.context.check = false
jonas.service.thread.file = jonas_areas.xml
jonas.service.discovery.ttl = 1
jonas.service.ha.datasource = jdbc_1
jonas.service.security.ws.realm = memrlm_1
jonas.service.ejb.minworkthreads = 3
jonas.service.dbm.class = org.objectweb.jonas.dbm.DataBaseServiceImpl
```



```

jonas.service.ws.wsgen.generator.factory = org.objectweb.jonas_ws.wsgen.generator.ews.EWSGeneratorFactory
jonas.service.ws.class = org.objectweb.jonas_ws.axis.AxisWSServiceImpl
jonas.service.resource.autoload = org.objectweb.jonas_ws.axis.AxisWSServiceImpl
jonas.service.jmx.class = org.objectweb.jonas.jmx.JmxServiceImpl
jonas.log.configfile = trace
jonas.service.ejb.maxworkthreads = 80
jonas.service.resource.minworkthreads = 5
jonas.service.web.autoload = org.objectweb.jonas_ws.axis.AxisWSServiceImpl
jonas.service.ha.timeout = 600
jonas.service.ha.gcl = jgroups
jonas.service.ear.parsingwithvalidation = true
jonas.service.ear.autoload = org.objectweb.jonas_ws.axis.AxisWSServiceImpl
jonas.service.resource.maxworkthreads = 80
jonas.service.ws.parsingwithvalidation = true
jonas.security.context.check.keystorePass = keystorepass
jonas.service.db.class = org.objectweb.jonas.db.hsqldb.HsqldbServiceImpl
jonas.service.ear.class = org.objectweb.jonas.ear.EarServiceImpl
jonas.service.jtm.class = org.objectweb.jonas.jtm.TransactionServiceImpl
jonas.service.ha.class = org.objectweb.jonas.ha.HaServiceImpl
jonas.service.ejb.threadwaittimeout = 60
jonas.service.ha.jgroups.groupname = jonas-rep
jonas.csiv2.propagation = true
jonas.service.ejb.class = org.objectweb.jonas.container.EJBServiceImpl
jonas.service.jms.topics = sampleTopic
jonas.service.jtm.timeout = 60
jonas.service.jms.collocated = true
jonas.service.security.csiv2.realm = memrlm_1
jonas.service.web.class = org.objectweb.jonas.web.wrapper.catalina55.CatalinaJWebContainerServiceWrapper
jonas.security.context.check.alias = FB
jonas.service.discovery.source.port = 9888
jonas.services = registry,jmx,jtm,db,dbm,security,resource,ejb,ws,web,ear
jonas.transaction.propagation = true

- Contents of 'HSQL1.properties':
jdbc.maxconpool = 100
jdbc.samplingperiod = 30
jdbc.connmaxage = 1440
jdbc.maxwaittime = 5
datasource.url = jdbc:hsqldb:hsqldb://localhost:9001/db_jonas
datasource.mapper = rdb.hsql
jdbc.minconpool = 10
jdbc.connteststmt = select 1
datasource.username = jonas
datasource.name = jdbc_1
datasource.classname = org.hsqldb.jdbcDriver
jdbc.maxwaiters = 100
datasource.password = jonas
jdbc.connchecklevel = 0
jdbc.maxopentime = 60

- Contents of 'trace.properties':
(file:/home/coqp/JONAS_4_8_5/conf/trace.properties)
handler.logtest.pattern = %d : %l : %h : %O{1}.%M : %m%n
logger.org.objectweb.jorm.level = WARN
logger.org.objectweb.speedo.level = WARN
logger.root.level = INFO
handler.mesonly.type = Console
logger.org.objectweb.jonas_lib.genbase.level = INFO
logger.org.objectweb.speedo.generation.SpeedoCompiler.level = INFO
logger.org.objectweb.jorm.generator.velocity.level = ERROR
logger.org.objectweb.jotm.level = INFO
logger.org.objectweb.medor.level = WARN
handler.wsdl.pattern = %d : %l : %h : %O{1}.%M : %m%n
handler.wsdl.output = jonas_wsdl.log
logger.org.objectweb.jonas_ws.wsgen.handler.0 = mesonly
logger.org.objectweb.jonas.genic.handler.0 = mesonly
handler.logtest.type = File
log.config.classname = org.objectweb.util.monolog.wrapper.javaLog.LoggerFactory
logger.org.objectweb.level = INFO
handler.tty.type = Console
handler.logf.output = automatic
logger.org.objectweb.jonas.genic.velocity.level = ERROR
logger.org.objectweb.jonas_tests.history.handler.0 = logtest
logger.org.objectweb.jonas.publication.additivity = false
logger.org.objectweb.jonas.publication.handler.0 = wsdl
handler.tty.output = Switch
handler.logtest.output = jonas_tests.log

```

```
handler.logf.pattern = %d : %l : %h : %O{1}.%M : %m%n
logger.org.objectweb.jonas_lib.genclientstub.level = INFO
handler.mesonly.pattern = %m%n
logger.org.objectweb.speedo.init.level = INFO
logger.fr.dyade.aaa.level = ERROR
logger.org.objectweb.jonas.genic.level = INFO
logger.org.objectweb.jonas_tests.history.level = INFO
logger.org.objectweb.jonas_ws.wsgen.additivity = false
handler.tty.pattern = %d : %O{1}.%M : %m%n
logger.org.mortbay.util.jmx.ModelMBeanImpl.level = ERROR
logger.org.objectweb.jonas_ws.wsgen.level = INFO
logger.org.objectweb.jonas.jdbc.sql.level = DEBUG
logger.org.objectweb.speedo.mapper.jorm-factory.class-properties.level = INFO
logger.org.objectweb.jonas_tests.history.additivity = false
handler.mesonly.output = Switch
logger.root.handler.1 = logf
logger.root.handler.0 = tty
handler.logf.type = File
logger.org.apache.struts.util.level = WARN
logger.org.objectweb.jonas.genic.additivity = false
handler.wsdl.type = File
logger.org.objectweb.carol.level = INFO
logger.org.objectweb.jonas_ejb.deployment.digester.level = FATAL
logger.org.jgroups.level = FATAL

- Contents of 'carol.properties':
(file:/home/coqp/JONAS_4_8_5/conf/carol.properties)
carol.jeremie.url = jrmi://localhost:2000
carol.irmi.interfaces.bind.single = false
carol.cmi.rr.factor = 100
carol.jrmp.url = rmi://localhost:1099
carol.iiop.server.sslport = 2003
carol.jndi.java.naming.factory.url.pkgs = org.objectweb.jonas.naming
carol.jvm.rmi.local.call = false
carol.irmi.server.port = 0
carol.jrmp.server.port = 0
carol.irmi.url = rmi://localhost:1098
carol.iiop.PortableRemoteObjectClass = org.objectweb.jonas_lib.naming.JacORBPRODelegate
carol.iiop.url = iiop://localhost:2001
carol.iiop.server.port = 0
carol.jrmp.interfaces.bind.single = false
carol.protocols = jrmp
carol.cmi.multicast.groupname = G1
carol.cmi.jgroups.conf = jgroups-cmi.xml
carol.cmi.stub.debug = false
carol.cmi.url = cmi://localhost:2002
carol.jvm.rmi.local.registry = false
carol.jeremie.server.port = 0

- Check 'jonas-realm.xml':
File is present.

- Check 'JORAM configuration':
Ok

The JOnAS environment seems correct.
```

---

# **jclient command**

---

## Name

jclient — start a *heavy* java client

## Synopsis

```
jclient [options] java-class [args]
```

## Description

The jclient command allows the user to easily start a "heavy" java client that will be able to reach beans in remote JOnAS servers and start distributed transactions.

It is not the J2EE compliant way to run a java client which is to use to package the java client in a J2EE container client (refer to Client Packaging).

## Options

- |               |  |
|---------------|--|
| -cp classpath | Add an additional classpath before running the java program.   |
| -security     | Set a security manager using the policy file in \$JONAS_BASE/conf/java.policy.<br>(Used for automatic stubs downloading) |

---

# GenIC command

---

## Name

GenIC — generates the container classes for EJBs

## Synopsis

```
GenIC [ Options ] <InputFileName>
```

## Description

The GenIC utility generates the container classes for JOnAS from the given Enterprise Java Beans.

The *InputFileName* is either the file name of an ejb-jar file or the file name of an XML deployment descriptor of beans.

The GenIC utility does the following :

1. generates the sources of the container classes for all the beans defined in the deployment descriptor,
2. compiles these classes via the java compiler,
3. generates stubs and skeletons for those remote objects via the rmi compiler, and
4. if the InputFile is an ejb-jar file, adds the generated classes in this ejb-jar file.

## Options

- |                           |  |
|---------------------------|--|
| -d directory              | Specifies the root directory of the class hierarchy.<br><br>This option can be used to specify a destination directory for the generated files.<br><br>If the <b>-d</b> option is not used, the package hierarchy of the target class is ignored and the generated files are placed in the current directory.<br><br>If the <i>InputFile</i> is an ejb-jar file, the generated classes are added to the ejb-jar file, unless the <b>-noaddinjar</b> option is set. |
| -invokecmd                | Invokes directly the java class corresponding to the java compiler.<br><br>This is useful on Windows in the event of a <i>CreateProcess Exception</i> (this occurs when the command line is too long).<br><br>In this case tools.jar must be visible in the CLASSPATH  |
| -javac <i>options</i>     | Specifies the java compiler name to use (javac by default).  |
| -javacopts <i>options</i> | Specifies the options to pass to the java compiler.  |
| -keepgenerated            | Do not immediately delete generated files.   |
| -noaddinjar               | If the <i>InputFile</i> is an ejb-jar file, do not add the generated classes to the ejb-jar file.  |
| -nocompile                | Do not compile the generated source files via the java and rmi compilers.  |
| -novalidation             | Remove xml validation during parsing.  |

- protocols      Comma-separated list of protocols (jrmf,iiop,cmi) for which stubs should be generated.  
Default is jrmf
- rmiopts *options*      Specifies the options to pass to the rmi compiler.
- verbose      Displays additional information about command execution.
- nofastrmic      Disable the use of fastrmic for stubs/ties generation.

## Example

GenIC -d ../../classes sb.xml      generates container classes of all the Enterprise JavaBeans defined in the sb.xml file. Classes are generated in the ../../classes directory adhering to the classes hierarchy.

GenIC sb.jar      generates container classes for all the Enterprise JavaBeans defined in the sb.jar file and adds the generated classes to this ejb-jar file.

## Environment

If *InputFile* is an XML deployment descriptor, the classpath must include the paths of the directories in which the Enterprise Bean's classes can be found, as well as the path of the directory specified by the -d option.

If *InputFile* is an ejb-jar file, the classpath must include the path of the directory specified by the -d option.

---

# JMS commands



---

## Name

JmsServer — Launches the JORAM Server with its default options.

## Synopsis

```
JmsServer
```

## Description

Launches the JORAM Server with its default options.

The JORAM server configuration file is `$JONAS_BASE/conf/a3servers.xml`

## Example

The **JmsServer** command is typically run in the background:

```
JmsServer &      on Unix,
```

```
start JmsServer  on Windows
```

---

## Name

`joram_raconfig` — Changes to the parameters (host, port, server id) in the JORAM configuration files.

## Synopsis

```
joram_raconfig [[-p <port>] [-h <host>] [-s <serverid>]]
```

## Description

The `joram_raconfig` tool aims to facilitate changes to the parameters (host, port, server id) in the JORAM configuration files.

JORAM relies on several configuration files: `a3servers.xml`, `joramAdmin.xml`, `ra.xml`. With `joram_raconfig`, all these configuration files are updated and thus the consistency is ensured.

Files modified:

- `$JONAS_BASE/conf/a3servers.xml`
- `$JONAS_BASE/conf/joramAdmin.xml`
- `$JONAS_BASE/rars/autoload/joram_for_jonas_ra.rar` in which the file `META_INF/ra.xml` is updated.

## Options

<code>-p port</code>	<code>port</code> : listening port of the JORAM server
<code>-h host</code>	<code>host</code> : IP address of the JORAM server
<code>-s serverid</code>	<code>serverid</code> : server id of the JORAM server

---

# **RAConfig command**

---

## Name

RAConfig — generates a JOnAS specific resource adapter configuration file or a resource adapter file

## Synopsis

```
RAConfig [ Options ] <InputFileName> [<OutputFileName>]
```

## Description

The **RAConfig** utility provides the capability to extract, create a JOnAS -specific resource adapter configuration file (*jonas-ra.xml*) from an *ra.xml* file, or create or update a resource adapter file

With this command it is possible to :

- extract a JOnAS -specific resource adapter configuration file (*jonas-ra.xml*) from an *ra.xml* file (See option **-path**)
- create a new JOnAS -specific resource adapter configuration file (*jonas-ra.xml*) from an *ra.xml* file (See option **-new**)
- create a resource adapter file (*.rar file*) from a *dataSource.properties* file (See option **-p**)
- update a resource adapter file with a *jonas-ra.xml* ((See option **-u**)

The *InputFileName* is the file name of a the resource adapter.

The *OutputFileName* is the file name of an output resource adapter. This parameter is used with the **-p**(required) or **-u**(optional) options.

## Options

**-?** or **-HELP** options                      Gives a summary of the options.

**-p** or **-Property**                              Specifies the name of the <database>.properties file to process. The  
database\_properties\_file\_name              result of this processing will be a *jonas-ra.xml* file that will update  
the /META-INF/*jonas-ra.xml* file in the output rar.

**-DM,-DS,-CP,-XA**                      These options are related to the option **-p** above.

They specify the rarlink value to configure **respectively DriverManager, DataSource,ConnectionPoolDataSource and XAConnection** If **-DM** is used, then the conversion will be a direct reflection of the values specified in the **-p** <database>.properties file. If any of the other values are specified, then the *jonas-ra.xml* created will reflect options from the **-p** <database>.properties file and the user must edit the file based on information from the database provider for the specified type of datasource. Each database provider may have different config properties that need to be set and will be included in the database provider's documentation.

**-PATH** *output directory*                      Specifies the directory name to place the extracted *jonas-ra.xml* file.

**-NEW**                      Don't extract *jonas-ra.xml* but create a new one. The default value is the System attribute of *java.io.tmpdir*.

- Update *inputname*      Specifies the name of the XML file to process. This file will update the /  
*META-INF/jonas-ra.xml* file in the rar. If this argument is used, it is the only  
argument executed.
  
- Rarlink rarlink      Specifies the jndi name of an rar file with which to link. This option can be  
used when this rar file will inherit all attributes associated with the specified  
jndi name. If this option is specified in the jonas-ra.xml file, it is the only file  
needed in the rar, and the ra.xml file will be processed from the rarlink file.
  
- SecurityFile      Specifies the security property file to process and add security infor-  
*security\_file\_to\_process*      mation to jonas-ra.xml. This will map the specified principal name  
to the user on the EIS system. The specified file must be in the fol-  
lowing form: *principal = user::password* . When used in conjunc-  
tion with the **-ENcrypt** option, then the resulting information will  
be encrypted in jonas-ra.xml.
  
- ENcrypt      Used with **-SecurityFile** to encrypt the specified passwords.
  
- Jndiname *jndiname*      This option is deprecated with 1.5 Resource Adapter.  
  
For 1.0 Resource Adapter, this specifies the JNDI name of the connection  
factory. This name corresponds to the name of the <jndi-name> element of  
the <jonas-resource> element in the JOnAS -specific deployment descriptor.  
This name is used by the resource service for registering in JNDI the con-  
nection factory corresponding to this resource adapter.
  
- NoValidation      Turn off the xml dtd/schema validation.

## Examples

- RAConfig -dm -p MySQL1      Generates a MySQL\_dm.rar file linked to JOnAS\_jdbcDM.rar  
\$JONAS\_ROOT/rars/autoload/  
JOnAS\_jdbcDM MySQL\_dm      the jonas-ra.xml file inserted is created with values coming from  
the ra.xml file of the JOnAS\_jdbcDM.rar and values from the  
MySQL1.properties file  
  
This rar file can then be deployed and will replace the configured  
MySQL1 datasource.
  
- RAConfig -path . XX.rar      Extract the jonas-ra.xml of XX.rar in the working directory
  
- RAConfig -u jonas-ra.xml      Updates/inserts the jonas-ra.xml file into the MyRA.rar file.  
MyRA.rar

---

# **newjb command**

---

## Name

newjb — builds a new JONAS\_BASE directory

## Synopsis

newjb

## Description

The newjb utility builds a new JONAS\_BASE directory that allows the conformance tests to be launched. At the start, the user must choose:

- the protocol among jrmp,iiop,cmi
- the database
- the web container among tomcat, jetty

The tool generates the configuration automatically.

The \$JONAS\_BASE variable must be set before launching the tool; it specifies the path to the new directory that will be built.

The \$HOME/jb.config/lib directory must be created before launching the tool. It can contain some specific user configuration (see below).

The tool relies on JOnAS 's ant tasks (\$JONAS\_ROOT/build-jb.xml) and thus builds a configuration compatible with the JOnAS version. First, a JONAS\_BASE with default values is built, and then the configuration files are modified with the values defined in the centralized configuration file of newjb (see below).

A default configuration file is provided in \$JONAS\_ROOT/build-jb.properties. It contains the variable parameters used by the tool, such as port number and database properties.

A user configuration can be set in the \$HOME/jb.config/conf/jonas-newjb.properties file. If this file is present, the parameters it contains will override the default parameters.

By default, only the HSQL database can be configured with this tool. For other databases, the specific drivers must be stored in the \$HOME/jb.config/lib directory before the run and the properties must be set in the \$HOME/jb.config/conf/jonas-newjb.properties file.

The default script (\$JONAS\_ROOT/build-jb.xml) and its configuration (\$JONAS\_ROOT/build-jb.properties) can be used as an example for creating a configuration tool corresponding to a user's specific requirements.