



Mdarad-Toolbox Environment Configuration

Destined To

MDARAD Toolbox Users

January 29, 2007

Table of Content

1. Audience.....	4
2. Introduction.....	4
3. Java.....	4
3.1. Version.....	4
3.2. Installation.....	4
3.3. Configuration.....	5
4. ANT 1.6.....	5
4.1. Version.....	5
4.2. Installation.....	5
4.3. Configuration.....	5
5. JBoss Application Server (JAS) Configuration.....	5
5.1. Version.....	5
5.2. Installation.....	5
5.3. Configuration.....	5
5.4. Libraries.....	6
5.5. Startup / Shutdown.....	6
6. Apache2 Configuration.....	6
6.1. Version Supported.....	6

6.2. Installation.....	6
6.3. Main configuration.....	7
6.4. Modules configuration.....	7
6.4.1 Tomcat Connector.....	7
6.4.2 Compression.....	9
6.4.3 SSL.....	9
6.5. Starting / Stopping Apache2.....	10
7. PoseidonUML.....	10
7.1. Version Supported.....	10
7.2. Installation.....	10
8. MySQL Server.....	11
8.1. Version Supported.....	11
8.2. Installation.....	11
8.3. Configuration.....	11

1 Audience

This document is aimed at all MDARAD Toolbox users. It contains a tutorial needed to configure the deployment servers required by the web application generated to work properly.

2 Introduction

This document contains the procedure to install the servers required for the web application to work properly. The first section presents the installation procedure for the Apache2 server. This server is required in order to deliver great performances for hosting static content (javascript, css, images, etc). It offers the possibility to compress text content, add SSL encryption and forward calls to the application server.

The second section explains how to configure the Jboss Application Server (JAS). The application server

is used for hosting the required servlets (struts, quartz, etc) and to manager the data sources accesses (EJB, Hibernate, etc).

3 Java

3.1 Version

The current tutorial is for the Java 6.x version.

3.2 Installation

To be completed...

3.3 Configuration

To be completed...

4 ANT 1.6

4.1 Version

The current tutorial is for the Ant 1.6.5 version.

4.2 Installation

To be completed...

4.3 Configuration

To be completed...

5 JBoss Application Server (JAS) Configuration

5.1 Version

The current tutorial is for JAS version 4.2.1.GA.

5.2 Installation

Download the JAS from the Jboss website : <http://labs.jboss.com/jbossas/downloads/>.

Extract the downloaded archive on your hard disk in the JBOSS_HOME folder (ex : /usr/local/jboss-4.2.1.GA or c:\dev\apps\jboss-4.2.1.GA)

5.3 Configuration

In the file JBOSS_HOME/server/default/ear-deployer.xml, look for and change the following lines :

```
<attribute name="Isolated">true</attribute>
<attribute name="CallByValue">true</attribute>
```

In the file `JBOSS_HOME/server/default/conf/jboss-service.xml` look for and change the following line in the mbean `org.jboss.naming.NamingService`:

```
<attribute name="CallByValue">true</attribute>
```

In the file `JBOSS_HOME/server/default/conf/jboss-minimal.xml` look for and change the following line in the mbean `org.jboss.naming.NamingService` :

```
<attribute name="CallByValue">true</attribute>
```

In the file `JBOSS_HOME/server/default/deploy/jboss-web.deployer/META-INF/jboss-service.xml` look for and change the following line :

```
<attribute name="UseJK">true</attribute>
```

In the file `JBOSS_HOME/server/default/deploy/jboss-web.deployer/server.xml`, comment out the following connectors if they are active:

```
<!-- <Connector port="8080" address="{jboss.bind.address}"
maxThreads="250" maxHttpHeaderSize="8192"
emptySessionPath="true" protocol="HTTP/1.1"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" /> -->
<!-- <Connector port="8443" protocol="HTTP/1.1"
maxThreads="150" scheme="https" secure="true" clientAuth="false"
keystoreFile="d:/dev/apps/jakarta-tomcat-5.0.28/conf/tomcat.keystore"
keystorePass="jarcatomcat" sslProtocol = "TLS" /> -->
```

In the same file, look for and change the following lines :

```
<Connector port="8009" address="{jboss.bind.address}" protocol="AJP/1.3"
emptySessionPath="true" enableLookups="false" redirectPort="8443" />
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="node1"> ...
</Engine>
...
```

5.4 Libraries

To enable the MySQL database, you must copy the `mysql-connector*.jar` from the `mdarad-genapp/templates/lib` folder to the `JBOSS_HOME/server/deploy/lib` folder.

5.5 Startup / Shutdown

Startup the JAS by using the `JBOSS_HOME/bin/run.sh` or `JBOSS_HOME/bin/run.bat` script. You can also shutdown the server by using the `JBOSS_HOME/bin/shutdown.sh --server=localhost` or `JBOSS_HOME/bin/shutdown.bat --server=localhost` command

6 Apache2 Configuration

6.1 Version Supported

The current tutorial is for Apache2 Web Server version 2.2.x.

6.2 Installation

Download the Apache2 web server from the Apache Foundation website :

<http://httpd.apache.org/download.cgi>. You can find the related documentation at this URL :

<http://httpd.apache.org/docs/2.2/>.

Follow the instructions to install the Apache2 Web Server on the target machine.

- For Windows (Vista), install the Win32 Binary including OpenSSL version, MSI installer.
 - For development Environment
 - Choose port 80 as service (we will change the port later)
 - Choose localhost as domain name and server name
- For Ubuntu (Debian), execute the following command :

```
sudo aptitude install apache2
```

6.3 Main configuration

6.3.1 Windows:

- Edit the APACHE_HOME/conf/httpd.conf and change the port to the port needed by the web application.(default is 80, for development purposes use 8080).

Edit the APACHE_HOME/conf/httpd.conf and add the following lines replacing the PROJECT_NAME tag by your project name and STATIC_CONTENT_PATH tag by the directory to the static path (usually it is d:/dev/projects/PROJECT_NAME/output/cumul/src/static) :

```
Alias /PROJECT_NAME "STATIC_CONTENT_PATH"

<Directory "STATIC_CONTENT_PATH">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

6.3.2 Linux:

- Edit the file APACHE_HOME/ports.conf and change the port to the port needed by the web application (default is 80, for development purposes use 8080).

- Edit the APACHE_HOME/mods-available/alias.conf and add the following lines replacing the PROJECT_NAME tag by your project name and STATIC_CONTENT_PATH tag by the directory to the static path (usually it is /var/www/PROJECT_NAME) :

```
Alias /PROJECT_NAME "STATIC_CONTENT_PATH"

<Directory "STATIC_CONTENT_PATH">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

- Finally, go to the STATIC_CONTENT_PATH folder and change the permissions to be 755 :

```
sudo chmod 755 *
```

6.4 Modules configuration

6.4.1 Tomcat Connector

6.4.1.1 For Windows

- Download the module from <http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/>. Select the platform, the version (1.2.26 and up) and download the so file (ex: [mod_jk-1.2.26-httpd-2.2.4.so](#))
- Copy the file in the APACHE_HOME/modules directory but rename the file (mod_jk.so)
- In APACHE_HOME/conf create a file name mod-jk.conf with the following content (replace PROJECT_NAME with project)

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile /var/log/apache2/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
```

```
# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
JkMount /PROJECT_NAME/* loadbalancer
JkUnMount /PROJECT_NAME/static/* loadbalancer

# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
#JkMountFile conf/uriworkermap.properties

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

- Create the file `APACHE_HOME/conf/workers.properties`. In this file insert the following content replacing `PROJECT_NAME` with your project name.

```
workers.tomcat_home=TOMCAT_DIR
workers.java_home=JAVA_HOME
ps=\

# Define worker 'example'
worker.list=example

# Set properties for worker 'example' (ajp13)
worker.example.type=ajp13
worker.example.host=localhost
worker.example.port=8009

worker.example.cachesize=10
worker.example.cache_timeout=600
worker.example.socket_keepalive=1
```



```
worker.example.recycle_timeout=300
```

- In the main configuration file (httpd.conf), add the following line at the end of the file:

```
# Include jk (redirecton to tomcat) config file
Include conf/mod-jk.conf
```

6.4.1.2 For Ubuntu (Debian)

- the mod_jk connector must be installed. To do so, run the following command :

```
sudo apt-get install libapache2-mod-jk
```

- From the APACHE_HOME folder, create the APACHE_HOME/conf folder. In this folder, create the file mod_jk.conf and insert the following content replacing the PROJECT_NAME tag with your project name :

```
# Load mod_jk module
# Specify the filename of the mod_jk lib
#LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile /var/log/apache2/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount your applications
JkMount /PROJECT_NAME/* loadbalancer
JkUnMount /PROJECT_NAME/static/* loadbalancer

# You can use external file for mount points.
# It will be checked for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkermap.properties

# Add shared memory.
```

```
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

- In the conf folder, create an empty file named uriworkermap.properties.
- Also, still in the conf folder, create a file named workers.properties and change the host name by your domain name.

```
# Define list of workers that will be used
# for mapping requests
# The configuration directives are valid
# for the mod_jk version 1.2.18 and later
#
worker.list=loadbalancer,status

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=localhost
worker.node1.type=ajp13
worker.node1.lbfactor=1
# worker.node1.connection_pool_size=10 (1)

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1

# Status worker for managing load balancer
worker.status.type=status
```

- In the main configuration file (apache2.conf), add the following line at the end of the file :

```
# Include jk (redirecton to tomcat) config file
Include conf/mod-jk.conf
```

6.4.2 Compression

6.4.2.1 Windows

- In httpd.conf, uncomment line:

```
#LoadModule _module modules/mod_deflate.so
```

- Create a file APACHE_HOME/conf/deflate.conf with the following content:

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain text/xml
    text/javascript text/css
    AddOutputFilterByType DEFLATE application/x-javascript
    BrowserMatch ^Mozilla/4 gzip-only-text/html
    BrowserMatch ^Mozilla/4\.0[678] no-gzip
    BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
    DeflateFilterNote Input input_info
    DeflateFilterNote Output output_info
    DeflateFilterNote Ratio ratio_info
    LogFormat "%r" %{output_info}n/%{input_info}n (%{ratio_info}n%)' deflate
</IfModule>
```

- In the main configuration file (httpd.conf), add the following line at the end of the file:

```
# Include deflate config file
Include conf/deflate.conf
```

6.4.2.2 For Ubuntu (Debian)

- The module for compression is mod_deflate. To enable the module run the following command :

```
a2enmod deflate
```

- From the APACHE_HOME/mods-enabled directory, run the following commands :

```
ln -s ../mods-available/deflate.conf deflate.conf
ln -s ../mods-available/deflate.load deflate.load
```

- Edit the APACHE_HOME/mods-available/deflate.conf file and change the content for the following lines :

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain text/xml
    text/javascript text/css
    AddOutputFilterByType DEFLATE application/x-javascript
    BrowserMatch ^Mozilla/4 gzip-only-text/html
    BrowserMatch ^Mozilla/4\.0[678] no-gzip
    BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
    DeflateFilterNote Input input_info
    DeflateFilterNote Output output_info
    DeflateFilterNote Ratio ratio_info
    LogFormat "%r" %{output_info}n/%{input_info}n (%{ratio_info}n%)' deflate
```

```
CustomLog /var/log/apache2/deflate_log_test deflate
</IfModule>
```

6.4.3 SSL

6.4.3.1 For Windows

- Add to the Windows PATH the path to the apache bin directory (i.e: APACHE_HOME/bin)
- From a command prompt, go to the APACHE_HOME/conf directory and launch the following commands:

```
openssl req -config openssl.cnf -new -out jarca-ssl.csr
openssl rsa -in privkey.pem -out my-server.key
openssl x509 -in jarca-ssl.csr -out jarca-ssl.cert -req -signkey jarca-ssl.key -days 365
mkdir ssl
move jarca-ssl.* ssl
```

- Then open the httpd.conf file and uncomment the following lines

```
LoadModule ssl_module modules/mod_ssl.so
...
Include conf/extra/httpd-ssl.conf
```

- Now open the APACHE_HOME/conf/extra/httpd-ssl.conf file and modify the lines to have these:

```
SSLCertificateFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/ssl/jarca-ssl.cert"
...
SSLCertificateKeyFile "C:/Program Files/Apache Software Foundation/Apache2.2/conf/ssl/jarca-ssl.key"
```

- If your SSL port is 8443 rather than 443 (or anything else) open the APACHE_HOME/conf/extra/httpd-ssl.conf file and modify the port of these lines to match yours:

```
Listen 8443
...
<VirtualHost _default_:8443>
...
ServerName localhost:8443
```

6.4.3.2 For Linux

- The module for SSL encryption is mod_ssl. To enable the module run the following command :

```
a2enmod ssl
```

- From the APACHE_HOME/sites-available, copy the default file to a file named to something relevant to your project (ex : jarca-ssl, etc). Then, from the

APACHE_HOME/sites-enabled, run the following command :

```
ln -s ../sites-available/<new-file-name> 001-<new-file-name>
```

- Edit the APACHE_HOME/sites-available/<new-file-name> file. Change the following lines :

```
NameVirtualHost *:443  
<VirtualHost *:443>
```

- Then, before the </VirtualHost> tag, insert the following lines :

```
SSLEngine on  
SSLCertificateFile /etc/apache2/ssl/apache.pem
```

- Edit the APACHE_HOME/ports.conf file and add the following lines to the end of the file. You can change the port number to any port of your choice (8443 for development usually):

```
Listen 443
```

- Finally, you must create the certificate needed for SSL. To do so, execute the following command by replacing APACHE_HOME by the apache root folder :

```
sudo apt-get install ssl-cert  
sudo mkdir APACHE_HOME/ssl  
sudo make-ssl-cert /usr/share/ssl-cert/ssleay.cnf APACHE_HOME/ssl/apache.pem
```

6.5 Starting / Stopping Apache2

6.5.1 For Windows

Simply use the icon in the tray icon to open the monitor and stop or start the service

6.5.2 For Linux

To start, stop or restart the server, run the following commands :

```
sudo /etc/init.d/apache2 start  
sudo /etc/init.d/apache2 stop  
sudo /etc/init.d/apache2 restart
```

Every time the configuration is changed, the Apache2 Web Server must be restart to enable those changes.

7 PoseidonUML

7.1 Version Supported

The current tutorial is for version 3.x because there are no community edition for PoseidonUML after the 3.x version.

7.2 Installation

Download the community edition from the Gentleware website archive downloads section :

<http://www.gentleware.com/downloadarchive.html#c1242>.

Unzip (for Linux) or install using the installer on windows.

8 MySQL Server

8.1 Version Supported

The current tutorial is for version 5.x and later because the version 4.x and before doesn't have integrity constraints implemented.

8.2 Installation

For Windows, download and use the installer from the last version of mysql.

For Ubuntu (Debian), use the following command to install the MySQL server :

```
sudo apt-get install mysql-server-5.0
```

8.3 Configuration

To avoid problem with case-sensitive environment, use the MySQL Administrator interface. Go to the Startup Parameters-->Advanced (tab)-->Various (section) and change the *Make table names case insensitive* to '0-Store as Created, Case Sensitive'