TAMEE Installation Guide

Introduction

This document describes the requirements and necessary steps to install and configure the Tissue Array Management and Evaluation Environment (TAMEE). TAMEE relies on the Authentication and Authorization System (AAS), which is also part of the distribution.

Requirements

Hardware Requirements

Server with at least 1.5 GB of main memory available to the applications, storage requirements depend on the number and size of section images stored in the database.

Software Requirements

- Java installation: JDK 1.5.x, JAVA_HOME has to point to the JDK installation
- MySQL 4.0.x with InnoDB support

The JBoss Application Server is bundled with the application, so no separate installation is needed.

Installation

Download tamee.tar.gz, open a shell (or command prompt) and unpack the compressed tar file to the directory of your choice. (Note: On Windows there seems to be a problem using WinZip 9.0. Some files are not extracted with their correct name. Use WinZip 10.0 or 7-Zip instead).
A directory “jboss” will be created. Change to that directory and enter all further commands from there.

MySQL Database Setup

This document assumes that MySQL 4.0.x is installed, running and listening on the default port 3306.

MySQL configuration

Edit the MySQL configuration file (/etc/my.cnf or <MySQL install folder>\my.ini) and add the following line to support the storage of BLOBs:

```
 set-variable = max_allowed_packet=200MB
```

Creation of databases

Invoke the MySQL client using a privileged account. Enter

```
 source sql/AAS.sql;
 source sql/TAMEE.sql;
```

This will create two accounts (usermn/usermn for AAS and tmadbuser/tmadbuser for TAMEE) and two databases (usermanagement and tma) together with all necessary tables. Change the passwords if required. Enter

```
 show databases;
```

You should see tma and usermanagement in the list of available databases, if both the databases were created successfully.

If MySQL is not running on the same server as JBoss add the appropriate privileges to grant JBoss access to the MySQL database:

```
GRANT ALL PRIVILEGES ON usermanagement.* TO usermn@<MySQL IP>
IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON tma.* TO tmadbuser@<MySQL IP> IDENTIFIED BY
'<password>';
```

JBoss Application Server Setup

TAMEE and AAS will be running in JBoss 4.0.1. JBoss configuration was already done for both the applications, what is left is to configure the JAVA_HOME environment variable and the data sources properly.

Configure the JAVA_HOME environment variable

`JAVA_HOME` has to point to a JDK 1.5.x installation on the server.

- For a Windows system use set JAVA_HOME=<JDK_folder> for a temporary definition or set it via “My Computer->Properties->Advanced->Evironment variables”.
- On a Unix system edit the file bin/run.conf and change the following line to match your configuration:

```
JAVA_HOME="/home/tma/java"
```

Edit the datasource configuration file for AAS

Open the file `server/aas/deploy/mysql-ds.xml`

Change the following entries to match your configuration

```
<connection-url>jdbc:mysql://10.1.0.12:3306/usermanagement
</connection-url>
<user-name>usermn</user-name>
<password>usermn</password>
```

Define mail relay for AAS

When a user is created, AAS sends an e-mail to the user specifying the username and the password. If a user is disabled a notification e-mail is sent to the administrator. Therefore the mail-server and a sender e-mail address have to be specified.

Open the file `server/aas/conf/UsermanagementEJB.properties` file and change the following lines to match your setup:

```
mailrelay=mailrelay.tugraz.at

#if a user gets disabled, the mail to the user has the sender:
from=Max.Muster@tugraz.at
```

Configure logging for AAS

To log application messages, AAS utilizes the log4j library. By default all log messages are written to a log file. Log behavior can be configured by editing the file `server/aas/conf/log4j.xml` and changing the following lines to match your local requirements:

```
<appender name="FILE"
class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="/var/log/j2ee/tma/server.log"/>
```

Edit the properties file for TAMEE

Open the file `server/default/conf/tma.properties`

Change the `protocol` entry to `http` and the `http_url` to the server IP in which the AAS application will be deployed. TAMEE uses this information to connect to the AAS system.

```
protocol=HTTP
##### HTTP Protocol #####
http_url=10.2.0.6
```

Change the `COOKIEDOMAIN` entry to reflect your domain name. If you access the server with the IP address or localhost disable the `COOKIEDOMAIN` entry by adding a # as the first character.

Edit the datasource configuration file for TAMEE

Open the file jboss/server/default/deploy/mysql-ds.xml

Change the following entries to match your configuration

```
<connection-url>jdbc:mysql://10.1.0.12:3306/tma</connection-url>
<user-name>tmadbuser</user-name>
<password>tmadbuser</password>
```

Configure logging for TAMEE

To log application messages, TAMEE utilizes the log4j library. By default all log messages are written to a log file. Log behavior can be configured by editing the file jboss/server/default/conf/log4j.xml and changing the following lines to match your requirements:

```
<appender name="FILE"
class="org.jboss.logging.appender.DailyRollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="/var/log/j2ee/tma/server.log"/>
```

Starting the Applications

Change to the JBoss binaries directory and enter all following commands from there:

```
cd bin
```

AAS

First start the Authentication and Authorization System

```
./run.sh –c aas       (any Unix system)
run.bat –c aas        (on Windows)
```

The application should be deployed without any errors. You should see a line similar to the following line at the end

```
INFO  [Server] JBoss (MX MicroKernel) [4.0.1 (build:
CVSTag=JBoss_4_0_1 date=200505022023)] Started in 45s:702ms
```

To check if everything is fine, open a browser and go to http://<server name or ip>:17080/UsermanagementWeb/

Login with username: `admin` and password: `12345`.

You should be able to login without any problem. Change the password immediately after login, since this is the administrator account.

TAMEE

First start the Authentication and Authorization System

```
./run.sh –c default  (any Unix system)
run.bat –c default   (on Windows)
```

The application should be deployed without any errors. You should see a line similar to the following line at the end

```
INFO  [Server] JBoss (MX MicroKernel) [4.0.1 (build:
CVSTag=JBoss_4_0_1 date=200505022023)] Started in 45s:702ms
```

To check if everything is fine, open a browser and go to http://<server name or ip>:18080/tma/

Login with username: `test` and password: `12345`. This account was created automatically during setup of the AAS and is assigned to the tmaAdmin group.
You should be able to login without any problem.

If all the above steps have been implemented, then you should be able to see the login page. Make sure that the AAS server is also running before attempting to login. After successful login, you can navigate through the side menu to different tables. Except the Dictionary table all the other tables should be empty. Click on the Dictionary menu to see the list of dictionary values.

Now the initial setup of both applications has been completed successfully and what is left is to customize TAMEE for everyday use as described in the additional steps below.

Additional Steps

Creating Users

Create a new user following Users-create and filling in the details. You should be able to create the user without any problem. JBoss console will show some error regarding the LDAP status. This can be safely ignored.

Adding Users to a Group

There are 3 predefined groups related to TAMEE: tmaAdmin, tmaUser, and tmaGuest. Assign the users created in the previous step to the appropriate group using the “groups” tab in the user details.

Creating Additional Groups and Access Control Lists (ACLs)

This step is optional and depends on the requirements of your organization. AAS facilitates the creation of additional groups and related ACLs. This is done as follows:
Create a new group via Group->create. Select “tma” as the application and enter the name of the group (prefix it with “tma” for consitency reasons).
Click on Acls->Create in the menu on the left side.
In the right side, choose “tma” as the application and the newly created group as the group.
Select the resource you want to add to that group and set the access rights.
Repeat the previous step for all resources you want the group give access to.

Additionally, you can change the existing ACLs using Acls->show/edit.

Customizing the Data Dictionary

The Data Dictionary is populated with a set of standard entries. Customized and extend the entries following Data Dictionary-Add New Item and Data-Dictionary->View all Items. Choose the Domain from the drop down list and add corresponding values to it.
For example, if ‘Microtome’ domain is selected, then add the following values. The value is important, description can be left empty.

Be careful while entering the DataDictionary values. It is very important to go through all the domains, list the corresponding potential values for each domain and make the dictionary entries one by one before you start using the system.

Applets

Gridding of section images and the two-color analysis is implemented via applets, which access the JBoss application server via the Java Naming and Directory Interface (JNDI and Remote Method Invocation (RMI). This requires that network ports used by JNDI and RMI are not blocked by any firewall in between the client and the server:

```
rmiactivation  18098/tcp      RMI Activation
rmiactivation  18098/udp      RMI Activation
rmiregistry    18099/tcp      RMI Registry
rmiregistry    18099/udp      RMI Registry
```

If you have any questions, please feel free to contact **TAMEE@genome.tugraz.at**

Appendix

Format of File for Bulk Definition of Tissues and Donors

The file for the bulk definition of tissues and donor blocks is tab delimited and consists of at least column. The first line contains the column names. Column order is not

Species	Species of the tissue (e.g. “Homo sapiens”) as defined in the data dictionary
Diagnose	Main diagnosis
Sample ID	Unique id of tissue sample (or case)
Topography	ICD-O coded topography of tissue (e.g. C25.9)
Morphology	ICD-O coded morphology of tissue (e.g. M8140)
Classification	Classification (“behavior”) of tissue (1..5)
Grade	Grade of tumor (1..5)
Paraffin ID	Donor block barcode
Core Comments	Additional comments for core

Format of File for Definition of the Array Layout

The array block layout file is tab delimited and consists of 14 columns, data in the columns with a star are required. Column order has to be strictly adhered to. There is no line with column names. The first line contains already information for a core of the array.

A*. Barcode of donor block (has to match an existing donor block barcode)
B. File name of arrayer image donor block
C. File name of microscope image of donor block
D. Translation of arrayer image vs microscope image x-direction [pixel]
E. Translation of arrayer image vs microscope image y-direction [pixel]
F. Rotation of arrayer image vs microscope
G. Scaling of arrayer image vs microscope image
H. Flag whether image has to be mirrored (0 = not mirrored, 1 = mirrored)
I*. Index of this punch in the donor block
J. Comment
K*. x-position of punch in the microscope image of the donor block [pixel]
L*. y-position of punch in the microscope image of the donor block [pixel]
M. Row of punch in array block
N. Column of punch in array block