|  |
| --- |
|  |
|  |
| Instructions for the author: Headings present in the document should not be deleted |
|  |
|  |
|  |
|  |
| Naming Service  Development Environment Setup Instructions |
|  |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Affiliation | Editor | Approver |
| Andraž Požar  Karin Rathsman | Cosylab  ESS |  |  |
|  |  |  |  |

Document revision history

|  |  |  |
| --- | --- | --- |
| Version | Reason for revision | Date |
| 1.0 | Initial version | 2014-01-07 |
| 2.0 | Switch to different technologies | 2014-04-08 |
| 3.0 | Major update | 2015-05-20 |
| 3.0.1 | Changed title to Naming Service | 2015-06-15 |
| 3.0.2 | Added instructions for customization of definitions | 2015-08-17 |

Table of Contents

1. Development Platform 5

2. Development Environment 5

2.1 Java 5

2.1.1 Java JRE 5

2.1.2 Java SDK 5

2.2 Git 5

2.3 Maven 5

2.3.1 Maven parent configuration 6

2.1 Postgresql 6

2.2 Wildfly 6

2.2.1 Installation 6

2.2.2 Create Management User. 7

2.2.3 Add Postgresql JDBC driver. 7

3. Check out the Projects 8

3.1 Check out the application from Bitbucket 8

4. Customise 8

4.1 Update form persistence 8

4.2 Authentication 8

4.3 About 9

4.4 Naming Convention 9

4.5 Index page 9

4.6 Theme 9

5. Postgresql Database Preparation 9

5.1 Start PostgreSQL database server 9

5.2 Create new database and database user 10

5.3 Security Domain Configuration 11

5.3.1 NamesRBAC Security Domain Configuration 11

5.3.2 NamesTest Security Domain Configuration 12

6. Build Projects 14

7. Deployment of the Application 14

1. Development Environment Setup 16

2. Prepare virtual machine 16

3. Download and Install CentOS 16

4. Settings 17

5. Prepare CentOS to be used as a virtual machine. 17

6. Install Guest Additions 18

# Development Platform

Default development platform for Naming Service is CentOS-7 installed as a virtual machine on VirtualBoxVM with OS X as host. Instructions how to set up the virtual machine can be found in appendix A. Other development platforms can also be used but this document is written to set up development environment on aforementioned platform.

# Development Environment

## Java

### Java JRE

Java 7 should be selected as default java. Check this with the following command in terminal:

sudo alternatives --config java

Java 7should be selected as default. If it is not, please select it.

### Java SDK

Install java software development kit with the following command:

sudo yum install java-1.7.0-openjdk-devel

Set the java environment variable by a adding a script file java.sh to /etc/profile.d/ containing the following line:

export JAVA\_HOME=/usr/lib/jvm/java

## Git

To install git type

sudo yum install git

## Maven

To install maven execute

sudo yum install maven

Then, set the M2\_HOME and M2 variables by adding a script (maven.sh) under /etc/profiles.d/ with the following lines

export M2\_HOME=/opt/apache-maven

export M2=$M2\_HOME/bin

(or add the lines to ~/.bashrc ) Restart your terminal.

### Maven parent configuration

The project object model of the application extends the ess-java-config pom.xml. To make it available in the maven repository, navigate to a suitable directory in which the repositories will be checked out (hereinafter referred as ~/git) and clone the ess-java-config repository from bitbucket

git clone https://bitbucket.org/europeanspallationsource/ess-java-config

Navigate to ~/git/ess-java-config and install the ess-java-config pom.xml into your maven repository with

mvn install

## Postgresql

See [www.postgresql.org/download/redhat](http://www.postgresql.org/download/redhat) for detailed instructions. To install Postgresql type:

sudo yum install postgresql-server

Carry out the post installation:

sudo postgresql-setup initdb

sudo systemctl enable postgresql.service

## Wildfly

(For detailed instructions please refer to <https://docs.jboss.org/author/display/WFLY8/Getting+Started+Guide>)

### Installation

Download Wildfly application from wildfly.org/downloads:

wget http://download.jboss.org/wildfly/8.2.0.Final/wildfly-8.2.0.Final.tar.gz

Extract to suitable location (e.g. /opt):

sudo tar xvfz wildfly-8.2.0.Final.tar.gz -C /opt

Create a link without version to make updates easy:

sudo ln –s /opt/wildfly-8.2.0.Final.tar.gz /opt/jboss

Set the JBOSS\_HOME variable by adding a script (jboss.sh) with the following line

export JBOSS\_HOME=/opt/jboss

(alternatively add a line to ~/.bashrc and /root/.bashrc ). Restart your terminal.

### Create Management User.

Create a management user:

sudo $JBOSS\_HOME/bin/add-user.sh

At the prompt, select Management User (option a) and follow the instructions but do not change the default ManagementRealm as realm.

### Add Postgresql JDBC driver.

Download the JDBC41 driver (current version is 9.4-1201) from <https://jdbc.postgresql.org/>

wget https://jdbc.postgresql.org/download/postgresql-9.4-1201.jdbc41.jar

Start the standalone configuration of the Wildfly server:

sudo $JBOSS\_HOME/bin/standalone.sh

Connect to the wildfly server using the command line interface management tool:

$JBOSS\_HOME/bin/jboss-cli.sh --connect

Authenticate as the management user and add the following two command lines at the prompt:

module add --name=org.postgresql --resources=/pathToDownloads/postgresql-9.4-1201.jdbc41.jar --dependencies=javax.api,javax.transaction.api

/subsystem=datasources/jdbc-driver=postgresql:add(driver-name=postgresql,driver-module-name=org.postgresql,driver-xa-datasource-class-name=org.postgresql.xa.PGXADataSource)

# Check out the Projects from Bitbucket

Navigate to ~/git and execute the following in terminal:

git clone https:// bitbucket.org/europeanspallationsource/naming-convention-tool.git

The cloned project includes all Naming Convention Tool sources. All projects under the root ~/git/naming-convention-tool are independent and configured to be build by maven:

**NamingConventionTool:** Contains the Naming Service, the main web-base application to manage naming convention of a particle accelerator.

**NamingConventionTool-JAXB**: Provides JAXB implementation of POJOs and Resource definitions used by NamingConventionTool webservice

**NamingConventionTool-Client**: The client api to access the NamingConventionTool webservice. It provides the ability to retrieve names data registered in NamingConventionTool.

# Customise

## Update form persistence

For development, or on the first deployment, uncomment the following three lines in resources/META-INF/persistence.xml under the main source folder ~/git/naming-convention-tool/NamingConventionTool/src/main/ to look like this:

<properties>

<property name="hibernate.hbm2ddl.auto" value="update"/>

</properties>

Then the database tables will be created on the first time the application is deployed. In addition, initial naming convention data that has been filled out in the excel file java/org/openepics/names/services/NamingDatabaseImport.xlsx (under the main source folder) will be loaded to the database on the first deployment.

## Authentication

The naming convention tool offers two alternative solutions for user authentication: Role Based Access (RBAC) and Test. The default choice is the RBAC authentication for ESS. Therefore, If you would like to use the alternative authentication you need to comment and uncomment two lines in webapp/WEB-INF/beans.xml under the main source folder to look like this

<!-- <class>org.openepics.names.services.SessionServiceRBAC</class> -->

<class>org.openepics.names.services.SessionServiceTest</class>

Do the same with webapp/WEB-INF/test/jboss-web.xml

<!-- <security-domain>namesRBAC.security-domain</security-domain> -->

<security-domain>namesTest.security-domain</security-domain>

## About

Please specify information (contact information, version number etc.) about the application in resources/messages.properties under the main source folder.

## Naming Convention

The ESS naming convention is selected as the default alternative. To implement an alternative naming convention, e.g. java/org/openepics/names/services/FribNamingConvention.java, you need to edit this file and specify the alternative in webapp/WEB-INF/beans.xml under the main source folder.

## Naming Convention Definition

Edit java/org/openpics/names/services/NamingConventionDefinition.java to update definitions used in GUIs.

## Index page

Please edit webapp/index.xhtml under the main source folder. Or add a new one and update the web.xml file accordingly.

## Theme

Replace the ESS logo /webapp/resources/css/brand/logo.png to change the brand logo in the application header.

Copy or modify the webapp/resources/css/themeEss.css and update webapp/resources/css/cssLayout.css accordingly.

Note: The ESS theme is based on ThemeRoller but it is not advisable to implement a full theme from ThemeRoller. Rather, replace the branding colours with find/replace all. Then use themeRoller (<http://jqueryui.com/themeroller/>) to generate icons with your branding colours.

# Postgresql Database Preparation

## Start PostgreSQL database server

To see whether the PostgreSQL server is up and running and ready to be used execute the following command:

sudo systemctl status postgresql.service

If it is not running you have to start it with

sudo systemctl start postgresql.service

## Create new database and database user

Log in to your PostgreSQL server with the following command:

sudo –iu postgres psql

You should now be in psql console. You should see something like this:

psql (9.4.10)

Type “help” for help

postgres=#

Create new database using the following command:

CREATE DATABASE discs\_names;

Crete new user and assign him all privileges for previously created database.

CREATE USER discs\_names WITH ENCRYPTED PASSWORD 'discs\_names';

GRANT ALL PRIVILEGES ON DATABASE discs\_names TO discs\_names;

Database is now ready to be populated with tables and data. Leave psql console by executing:

\q

Open PostgreSQL configuration file with root permissions:

sudo gedit /var/lib/pgsql/data/pg\_hba.conf

Modify line

host all all 127.0.0.1/32 peer

to

host all all 127.0.0.1/32 md5

Restart the database:

sudo systemctl restart postgresql.service

Then, stop the Wildfly server with the command line interface management tool

$JBOSS\_HOME/bin/jboss-cli.sh --connect –command=:shutdown

and open $JBOSS\_HOME/standalone/configuration/standalone.xml. Under

<server>

<profile>

<subsystem xmlns="urn:jboss:domain:datasources:2.0">

add the data source used by Naming Service:

<datasource jta="false" jndi-name="java:/org.openepics.names.data" pool-name="java:/org.openepics.conf.dbpool" enabled="true" use-ccm="false">

<connection-url>jdbc:postgresql://localhost:5432/discs\_names</connection-url>

<driver-class> org.postgresql.Driver </driver-class>

<driver>postgresql</driver>

<security>

<user-name>discs\_names</user-name>

<password>discs\_names</password>

</security>

<validation>

<validate-on-match>false</validate-on-match>

<background-validation>false</background-validation>

</validation>

<statement>

<share-prepared-statements>false</share-prepared-statements>

</statement>

</datasource>

## Security Domain Configuration

### NamesRBAC Security Domain Configuration

Open $JBOSS\_HOME/standalone/configuration/standalone.xml file and add the following lines to configure the rbac properties:

<system-properties>

<property name="rbac.primaryServicesURL" value="https://rbac.esss.lu.se:8443/service"/>

<property name="rbac.primarySSLHost" value="rbac.esss.lu.se"/>

<property name="rbac.primarySSLPort" value="8443"/>

</system-properties>

where all the values needs to be customized to your rbac server configuration. In the same file, under

<server>

<profile>

<subsystem xmlns="urn:jboss:domain:security:1.2">

<security-domains>

add the namesRBAC security domain used by Naming Service:

<security-domain name="namesRBAC.security-domain" cache-type="default">

<authentication>

<login-module code="se.esss.ics.rbac.loginmodules.RBACLoginModule" flag="required">

<module-option name="RBACResource" value="NamingService"/>

<module-option name="RBACPermissionsOfInterest" value="Edit,Manage"/>

<module-option name="mapRBACPermissionsToASRoles" value="false"/>

</login-module>

</authentication>

</security-domain>

### NamesTest Security Domain Configuration

Authentication for the Test option is handled on the Wildfly application server. Therefore we must add all the users we want to have in the application to the application server and assign them roles. To this a small script has been created to minimize the effort put into creating new users and their passwords.

Move to appropriate directory (~/git) and clone a project from Bitbucket with which you will be able to generate passwords for users:

git clone https://*username*@bitbucket.org/ess\_ics/jboss-password-tool.git

Please substitute username with your user account on Bitbucket.

Move to newly created directory and execute:

mvn package

to build the project.

We will create two <username, password> pairs. One user will be later on assigned role of Naming Administrator and other role of Naming Editor.

To generate new username, password pair where username is “nameseditor” and password is also “nameseditor” execute the following:

./run nameseditor org.openepics.names.realm

when prompt for password enter “nameseditor” and then enter this again for confirmation.

<username, password> pair will be generated and it should look **exactly** like this:

nameseditor=d8519be29cbe13ac4d7d6b0b9706a3fd

Do the same for user “namesadmin” with password “namesadmin”. Generated <username, password> pair should look **exactly** like this:

namesadmin=0c29fb8b288ea8c86b60ae3f52d4a5da

Thereafter, create a file named names-users.properites under the $JBOSS\_HOME/standalone/configuration/ directory and add the two generated <username, password> pairs. When finished the content of $JBOSS\_HOME/standalone/configuration/names-users.properties should look exactly like this:

nameseditor=d8519be29cbe13ac4d7d6b0b9706a3fd

namesadmin=0c29fb8b288ea8c86b60ae3f52d4a5da

We now have usernames and passwords but we also have to set realm roles for those users. These roles are not used for authorization of actions in the applications. Those roles will be set further on. To set the roles create a file named names-roles.properties under the $JBOSS\_HOME/standalone/configuration/ directory and add the following two lines:

nameseditor=USER

namesadmin=USER

Next step is to configure the application server. Open $JBOSS\_HOME/standalone/configuration/standalone.xml file with your preferred text editor. Under

<server>

<profile>

<subsystem xmlns="urn:jboss:domain:security:1.2">

<security-domains>

add the following lines to configure the namesTest security domain:

<security-domain name="namesTest.security-domain" cache-type="default">

<authentication>

<login-module code="RealmUsersRoles" flag="required">

<module-option name="usersProperties" value="${jboss.server.config.dir}/names-users.properties"/>

<module-option name="rolesProperties" value="${jboss.server.config.dir}/names-roles.properties"/>

<module-option name="realm" value="org.openepics.names.realm"/>

<module-option name="password-stacking" value="useFirstPass"/>

</login-module>

</authentication>

</security-domain>

In order for the created users to be also recognized by the application we have to add them to application database and assign them application roles. The two users namesadmin and nameseditor will be created on the first deployment of the application, so you do not need to add them. Only if you follow the steps above to add other users, e.g. Alice with role EDITOR and Bob with role SUPERUSER you need to do execute the following set of commands:

sudo –u postgres psql –d discs\_names

to connect to “discs\_names” database. When connected:

INSERT INTO useraccount (version,role,username) VALUES (0,’SUPERUSER’,’Bob’);

INSERT INTO useraccount (version,role,username) VALUES (0,’EDITOR’,’Alice’);

and

\q

to exit the psql console.

# Build Projects

You can build the Naming Convention Tool code either in each individual project or by navigating to the root project ~git/naming-convention-tool and execute

mvn install -P test

The root pom.xml contains references to all Naming Convention Tool projects and will build and install them in proper order. The output of the build is located in the target folder of each individual project.

The profile “test” indicates that the application is build to be deployed in the development environment, i.e. on localhost:8080/names-test. For production the production profile (-P production) should be used. However, this is not in the scope of this manual.

# Deployment of the Application

Start the Wildfly server by executing the following command:

sh $JBOSS\_HOME/bin/standalone.sh

Add the war file you built with maven, currently version 3.2.1 as a link to the jboss deployment folder:

sudo ln -s ~/git/naming-convention-tool/NamingConventionTool/target/names-3.2.1.war $JBOSS\_HOME/standalone/deployments/

Now you can go to <http://localhost:8080/names>-test and login.

Appendix A

# Development Environment Setup

NamingService has been developed on CentOS 7.0 installed as a virtual machine on a host with the following specifications:

* Model: MacBook Pro
* Operating system: OS X Yosemite 10.10.2
* Processor: 2.6 GHZ Intel Core i7
* Memory: 16 GB 1600 MHz DDR3
* Graphics: NVIDIA GeForce GT 750 M 2048 MB

# Prepare virtual machine

Download and install Virtual Box from <https://www.virtualbox.org/wiki/Downloads>

1. Start VirtualBox
2. Click on "New" button in the menu bar of the VirtualBox Manager and enter the following settings:
   1. Name and operating system:
   2. Enter a name E.g. “CentOS-7” for the virtual machine.
   3. Select type "Linux"
   4. Select version "Red Hat (64 bit)"
3. Memory Size: Select 8192 MB RAM Memory
4. Hard drive: Select “Create new hard disk now” and press “Create”
5. Hard drive file type: Select “VMDK (Virtual Machine Disk) as hard drive file type
6. Storage on physical hard drive: Select “Fixed size”
7. File location and size:
   1. Select name and location for the virtual machine (Default location is ~/VirtualBox VMs/)
   2. Select 32 GB memory for virtual hard drive.
8. Check that the new virtual machine is visible in the VirtualBox Manager.

# Download and Install CentOS

1. Download the iso image ( CentOS-7-x86\_64-Everything-1503-01.iso ) from <https://www.centos.org/download/>
2. Select the new virtual machine in the VirtualBox Manager and press "Start"
   1. Press the folder icon to browse the host system and select the downloaded CentOS iso image.
   2. Press “Start” and select language.
3. In Installation Summary:
   1. Configure the environment. Do not select the minimum installation rather, select a workstation.
   2. Press install.
4. In User Settings: Set Root password and create a user with administrator privileges.

# Settings

1. From the VM VirtualBox Manager navigate to Settings /General/Advanced and select bidirectional for Shared Clipboard and Drag’n Drop
2. Navigate to Settings/Shared Folders/ Add a new share folder by clicking the add folder button, then Select Folder Path on your host system and name it e.g. Host. Do not check the Auto-mount option.
3. To avoid frustration for Mac users: From the VM VirtualBox Manager, Select VirtualBox VM>Preferences… from the menu bar and navigate to input and the Virtual Machine tab. Klick the Shortcut for the Host Key Combination and replace left cmd key with e.g. right cmd key.

# Prepare CentOS to be used as a virtual machine.

1. Start the new virtual machine if it is not already running. Note: Until the guest additions have been installed the virtual machine captures the pointer. To switch focus you need to press the host key (default is left cmd on OS X host and right ctrl on Windows hosts).
2. Login
3. Switch off Screen Lock and screen saver:
   1. Select Application>System Tools> Settings from the menu bar. Select Privacy and switch the screen lock off.
   2. navigate back to settings and select Power. Set Power Saving to Never.
4. Establish network connection
   1. Select Application>System Tools>Settings the menu bar, select Network and switch on Wired. (Alternatively, switch on the network from the network icon on the desktop toolbar)
5. Open a terminal and type

sudo yum update

1. Install the GNU C compiler and the kernel development packages:

sudo yum install epel-release

sudo yum install dkms

sudo sudo yum install gcc

sudo yum install kernel-devel

1. Reboot your guest system in order to activate the updates

sudo shutdown -r now

1. Mount shared folder
   1. In a terminal, create a folder ~/host
   2. Add the following line to the /etc/fstab file:

Host /home/<username>/host vboxsf rw,noauto,noexec,uid=1000,gid=1000 0 0

* 1. Auto mount does not work, so to mount the shared folder you just type

sudo mount ~/host

* 1. Alternatively: mount the created shared folder Host with

sudo mount -t vboxsf -o rw,uid=1000,gid=1000 Host ~/host

# Install Guest Additions

1. Select Devices>Insert Guest Additions CD Image.. (Host D) from the virtual machine’s menu bar. (Alternatively: Insert the VBoxGuestAdditions.iso CD file into the CentOS virtual CD-ROM drive. On Mac OS X hosts, you can find this file in the application bundle of VirtualBox. (Right click on the VirtualBox icon in Finder and choose Show Package Contents. It is located in the Contents/MacOS folder. Copy this file to a location on your host file system. Then, from the from the virtual machine’s menu bar, select Devices> CD/DVD-Devices> Chose a virtual CD/DVD disk file… and select the ISO image.)
2. Execute VBoxGuestAdditions either by a double click on the CD/DVD image on the desktop (might not work) or navigate to the default media folder (usually /media or run/media/<username>/) in a terminal and type

sudo sh VBOXADDITIONS\_<version>/VBoxLinuxAdditions.run

1. Reboot your guest system if necessary