



Installation Guide

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. <http://creativecommons.org/licenses/by-nc-sa/3.0>



Contents

1	Development Environment	4
1.1	Prerequisites	4
1.1.1	Java	4
1.1.2	Ruby	4
1.1.3	MySQL	4
1.1.4	Openfire	5
1.1.5	WireShark	5
1.2	Post Installation Steps	5
1.2.1	Set environment variables	5
1.3	Directory Layout	5
1.4	Environment Settings	5
1.4.1	SERVER_HOME	6
1.4.2	DEV_HOME	6
1.4.3	TINOS_HOME	6
1.5	Directory Setup	7
1.6	Install Virgo WebServer	7
1.6.1	Installation	8

1.6.2	Environment Settings	8
1.7	Ant	8
1.7.1	Installation	8
1.7.2	Environment Settings	9
1.8	Findbugs	9
1.8.1	Installation	9
1.8.2	Environment Settings	9
1.9	SpringSource Tool Suite	10
1.9.1	Environment Settings	10
1.10	Overall Environment Settings	10
1.11	Openfire Database Installation	11
1.12	OpenFire 3.6.4	12
1.13	OpenFire Configuration	12
1.14	OpenFire Demo Configuration	19
1.15	Virgo Server Configuration	21
1.16	XMPP/Jabber IM Client	21
1.17	TINOS - Simple Ping Demo	23
1.17.1	Initial Run of the Demo	24
1.17.2	Stopping the Virgo Server	25
1.17.3	Post-Initial Demo	25
1.18	Useful Links	25

Chapter 1

Development Environment

1.1 Prerequisites

1.1.1 Java

Java SE 6 or later is required. The setup and installation of Java SE 6 is typically operating system specific, so consult your OS provider.

1.1.2 Ruby

The Ruby scripting language is required for the source project generation templates. The setup and installation of Ruby is typically operating system specific, so consult your OS provider. Additional ruby plugins (gems) are also required to support the template system (choice, xml-simple).

1.1.3 MySQL

MySQL is required to support the installation of the Openfire XMPP server. Download and Installation information is available at :

<http://www.mysql.com>

1.1.4 Openfire

The Openfire XMPP server is required to support the construction of virtual networks within the TINOS platform. Download and Installation information is available at :

<http://www.igniterealtime.org/projects/openfire>

1.1.5 WireShark

WireShark is a tool that is used to examine TINOS network trace logs. Download and installation information is available at :

<http://www.wireshark.org>

1.2 Post Installation Steps

1.2.1 Set environment variables

JAVA_HOME

TINOS uses the JAVA_HOME environment variable to locate the java executable. Configure this environment variable to point to the home directory of the Java 5 or 6 installation on your computer.

1.3 Directory Layout

1.4 Environment Settings

Generally we attempt to isolate environments from each other as a matter of course. This supports the idea of multiple environments being available for a user dependent upon what they want to do.

To support this idea, all the environment settings for TINOS development are written into the file `env.sh` in the development directory. This file can be sourced within the shell of the user to append the relevant settings to their default shell.

In doing so, the user tailors this shell for the TINOS development environment and platform. Listings of the relevant settings under the various headings of the installation are provided just to help join the dots for the reader. A complete `env.sh` file is also listed at the end of the document.

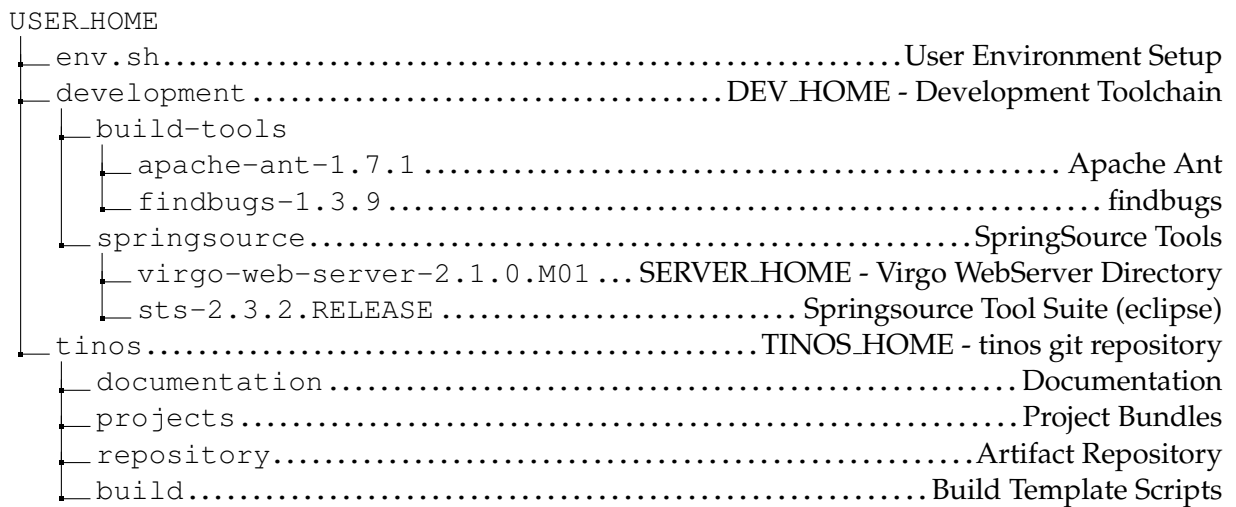


Figure 1.1: High Level Installation Layout

1.4.1 SERVER_HOME

As a convenience it is recommended that you create an environment variable that points to the Virgo Web Server installation directory. Note that the Virgo Web Server does not required that such an environment variable has been set. This variable may have any name of your choosing. The following documentation assumes that the variable is named SERVER_HOME.

1.4.2 DEV_HOME

As a convenience it is recommended that you create an environment variable that points to the directory that contains the development environment tooling. This variable may have any name of your choosing. The following documentation assumes that the variable is named DEV_HOME.

The location of the DEV_HOME directory is typically "development" located in the user home directory. However, it is up to the user to choose this location.

1.4.3 TINOS_HOME

As a convenience it is recommended that you create an environment variable that points to the directory that contains the tinos git repository. The following documentation assumes that the variable is named TINOS_HOME.

The location of the TINOS_HOME directory is typically "tinos" located in the user home directory. It is up to the user to choose this location.

1.5 Directory Setup

Task Setup the development toolchain.

```
$ mkdir development; cd development
```

```
$ export DEV_HOME='pwd'
```

```
$ mkdir build-tools
```

```
$ mkdir springsource
```

```
$ echo $DEV_HOME
```

Note Copy down the value of DEV_HOME as it will need to be added to the env.sh file.

Task Pull down the TINOS repository from GitHub

```
$ cd
```

```
$ git clone $ GitHub Repository URL for TINOS $
```

```
$ cd tinos
```

```
$ export TINOS_HOME='pwd'
```

```
$ echo $TINOS_HOME
```

Note Copy down the value of TINOS_HOME as it will need to be added to the env.sh file.

1.6 Install Virgo WebServer

The current version of Virgo is virgo-web-server-2.1.0.M01.

TINOS is loaded and executed within the Virgo WebServer platform. It is recommended that the Virgo WebServer is installed in the DEV_HOME directory as TINOS users will need to interact with this server directly in order to load and execute TINOS nodes. Download and installation information is available at :

<http://www.eclipse.org/virgo>

1.6.1 Installation

Task Download the Virgo Web Server from Eclipse.

URL <http://www.eclipse.org/virgo/download>

Note Assuming the zip file is downloaded to the HOME directory.

```
$ cd $DEV_HOME/springsource
```

```
$ unzip $HOME/virgo-web-server-2.1.M01.zip
```

Note This will extract the virgo server into virgo-web-server-2.1.0.M01 directory.

```
$ cd virgo-web-server-2.1.0.M01/bin
```

Note Create a file called setenv.sh and place the following in it.

```
# Workaround (OSX)
# Java JVM issue in relation to the churn of the PERM cache.
export JAVA_OPTS="-Xms64m -Xmx512m -XX:PermSize=128m -XX:MaxPermSize=756m"
```

1.6.2 Environment Settings

```
# Virgo Settings
export SERVER_HOME=$DEV_HOME/springsource/virgo-web-server-2.1.0.M01
export SERVER_EXEC=$SERVER_HOME/bin
```

1.7 Ant

The current version of Ant is apache-ant-1.7.1.

1.7.1 Installation

Task Download the Apache Ant from Apache.Org.

URL <http://ant.apache.org/bindownload.cgi>

Note Assuming the zip file is downloaded to the HOME directory.

```
$ cd $DEV_HOME/build-tools
```

```
$ unzip $HOME/apache-ant-1.7.1-bin.zip
```

Note This will extract the ant into the directory: apache-ant-1.7.1

1.7.2 Environment Settings

```
# Ant Settings
export ANT_HOME=$DEV_HOME/build-tools/apache-ant-1.7.1
# Workaround (OSX)
# Java JVM issue in relation to the churn of the PERM cache.
export ANT_OPTS="-Xms64m -Xmx512m -XX:PermSize=128m -XX:MaxPermSize=756m"
export ANT_EXEC=$ANT_HOME/bin
```

1.8 Findbugs

The current version of Findbugs is findbugs-1.3.9.

1.8.1 Installation

Task Download the findbugs from

URL <http://findbugs.sourceforge.net/downloads.html>

Note Assuming the zip file is downloaded to the HOME directory.

```
$ cd $DEV_HOME/build-tools
```

```
$ tar xzvf $HOME/findbugs-1.3.9.tgz
```

Note This will extract the findbugs into the directory: findbugs-1.3.9

1.8.2 Environment Settings

```
export FINDBUGS_HOME=$DEV_HOME/build-tools/findbugs-1.3.9
export FINDBUGS_EXEC=$FINDBUGS_HOME/bin
```

1.9 SpringSource Tool Suite

Install the Spring Tool Suite (SpringSource branded version of *Eclipse*), it is invoked on the command line as "STS".

Note: The GUI toolchain (STS/Eclipse) will be updated shortly to handle the upgrade changes from Spring dm to Virgo. Command line operation is not affected, it is only the integration with GUI development tools. So expect changes here.

Task Download SpringSource Tool Suite.

URL <http://www.springsource.com/products/springsource-tool-suite-download>

```
$ cd $DEV_HOME/springsource
```

```
$ unzip $HOME/sts-2.3.2.RELEASE.zip
```

Note This will extract into the following directory : sts-2.3.2.RELEASE

1.9.1 Environment Settings

```
export STS_HOME=$DEV_HOME/springsource/sts-2.3.2.RELEASE
```

1.10 Overall Environment Settings

All of the environmental settings from above combined and integrated. Typically this file is imported into the users shell whenever they wish to use the environment.

Task Append the TINOS Environment settings, a sample env.sh file is shown below.

```
$ cd
```

```
$ ../env.sh
```

Joy Ready to rock and roll!.

```
# Development Toolchain Location
export DEV_HOME=$HOME/development

# TINOS Git Repository
export TINOS_HOME=$HOME/github/tinos

# Java JDK/JRE (OSX)
export JAVA_HOME=/Library/Java/Home

# Virgo
export SERVER_HOME=$DEV_HOME/springsource/virgo-web-server-2.1.0.M01
export SERVER_EXEC=$SERVER_HOME/bin

# STS
export STS_HOME=$DEV_HOME/springsource/sts-2.3.2.RELEASE

# Ruby Settings
export RUBYOPT=rubygems

# Ant Settings
export ANT_HOME=$DEV_HOME/build-tools/apache-ant-1.7.1
# Workaround (OSX)
# Java JVM issue in relation to the churn of the PERM cache.
export ANT_OPTS="-Xms64m -Xmx512m -XX:PermSize=128m -XX:MaxPermSize=756m"
export ANT_EXEC=$ANT_HOME/bin

# Findbugs
export FINDBUGS_HOME=$DEV_HOME/build-tools/findbugs-1.3.9
export FINDBUGS_EXEC=$FINDBUGS_HOME/bin

# Setup Path
export PATH=$JAVA_HOME/bin:$ANT_EXEC:$SERVER_EXEC:$PATH
export PATH=$STS_HOME:$FINDBUGS_HOME:$PATH

# Alias
export EDITOR=vim
alias vi='vim'
```

1.11 Openfire Database Installation

Before the installation of Openfire, a database user and database for the Openfire server must be created.

Task Create database user and database

Note Create a database

```
$ mysql -u root -p
```

```
mysql$ create database openfire character set utf8;
```

Note Create a user

```
mysql$ grant all on openfire.* to openfire@localhost identified by 'openfire';
```

```
mysql$ commit; exit;
```

1.12 OpenFire 3.6.4

Install the OpenFire server as instructed on their website. Once this is completed, start the server and then follow the instructions below to configure the server.

1.13 OpenFire Configuration

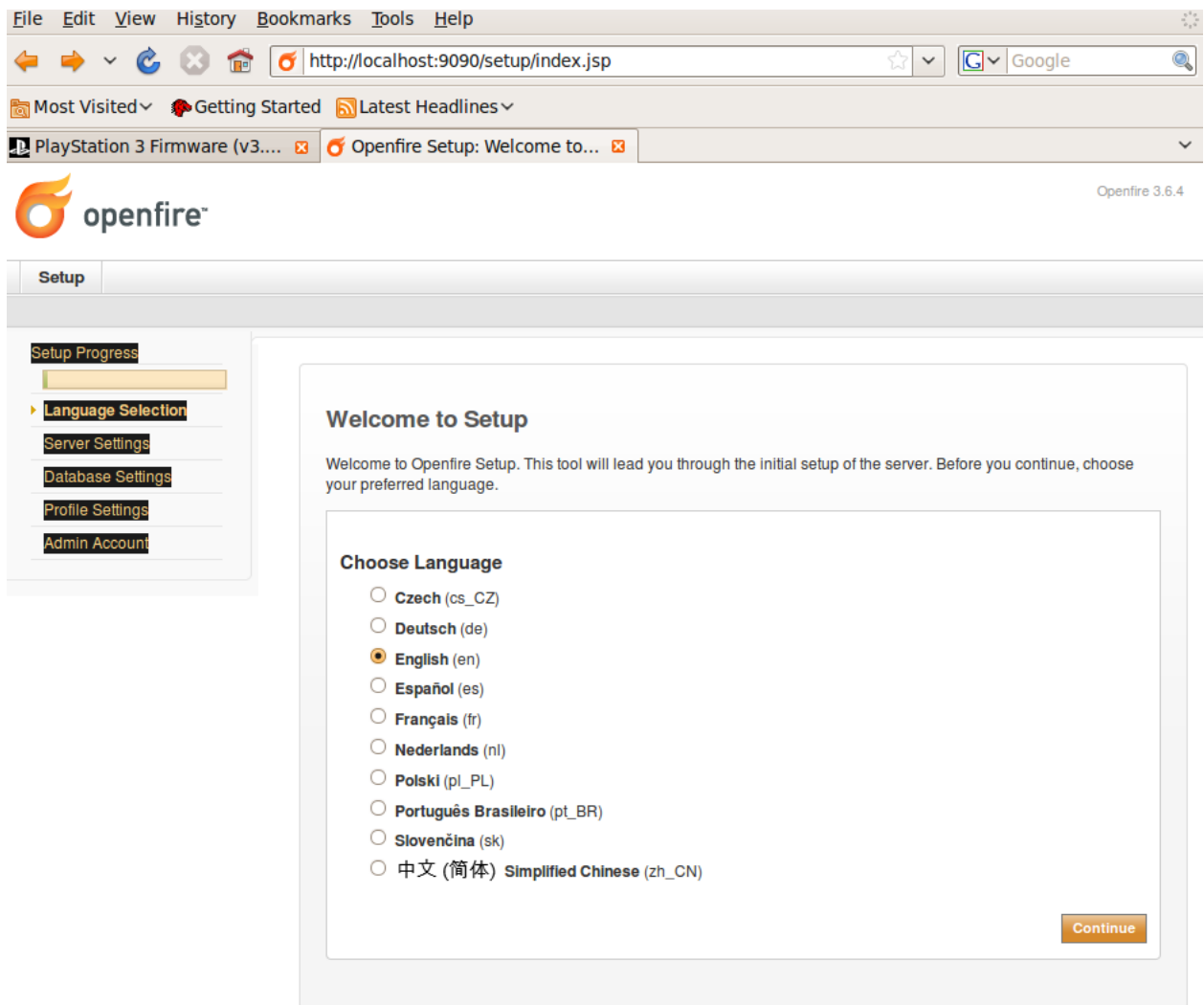
In order to complete the configuration of the OpenFire server, a web browser must be used to step through the server setup screens.

Task Configure the OpenFire server with a web browser.

Browser Enter the following URL to start the configuration.

URL <http://localhost:9090>

Browser Select the English language and click Continue.



Built by [Jive Software](#) and the [IgniteRealtime.org](#) community

Task Configure the Server Settings

Browser Enter "localhost" as the domain and leave the other settings unchanged.

Browser Click Continue

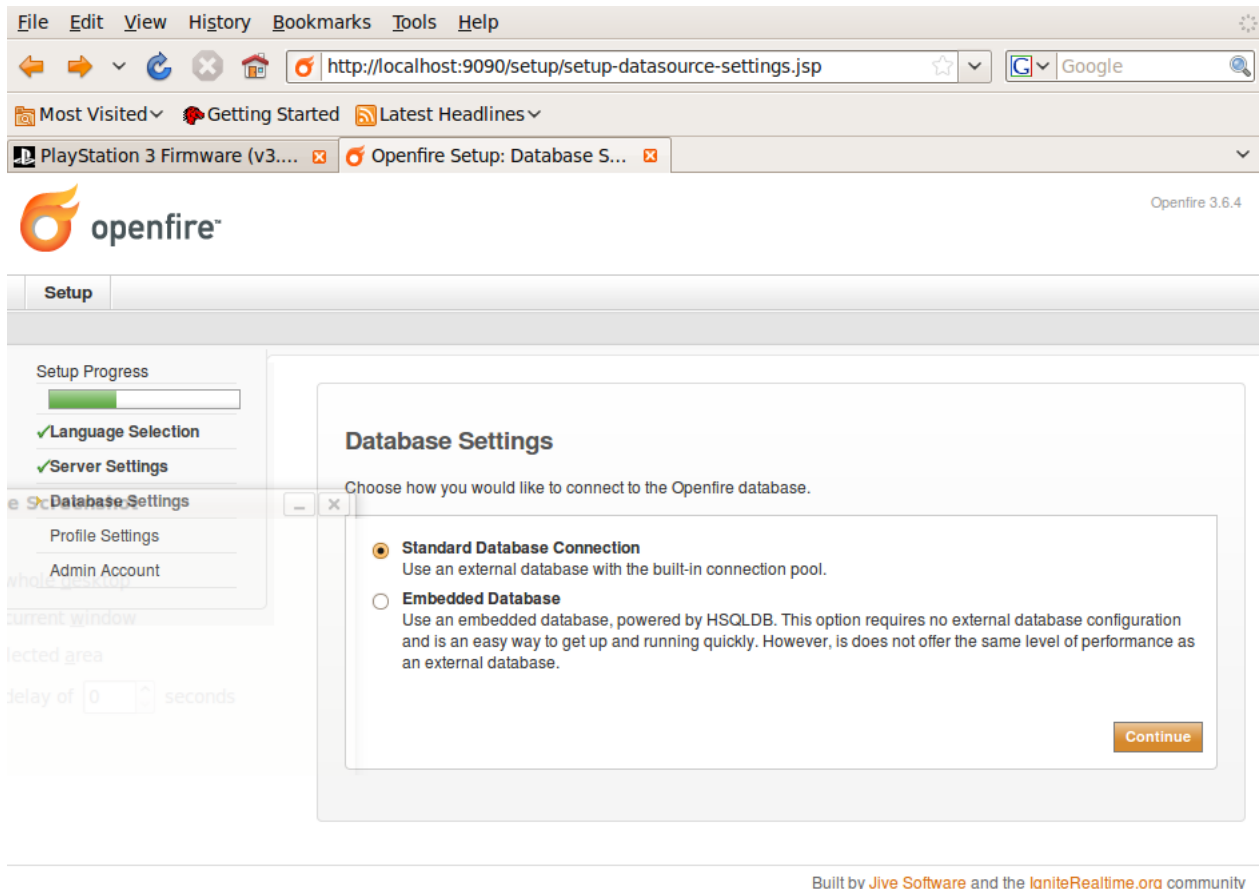
The screenshot shows a web browser window with the URL `http://localhost:9090/setup/setup-host-settings.jsp`. The browser's address bar and tabs are visible. The Openfire logo and version (3.6.4) are at the top left. The main content area is titled "Server Settings" and contains a note: "Below are host settings for this server. Note: the suggested value for the domain is based on the network settings of this machine." Below the note are three input fields: "Domain:" with the value "localhost", "Admin Console Port:" with the value "9090", and "Secure Admin Console Port:" with the value "9091". Each field has a help icon (question mark). A "Continue" button is located at the bottom right of the form. On the left side, there is a "Setup Progress" section with a progress bar and a list of steps: "Language Selection" (checked), "Server Settings" (active), "Database Settings", "Profile Settings", and "Admin Account".

Built by [Jive Software](#) and the [IgniteRealtime.org](#) community

Task Configure the Database Settings

Browser Select "Standard Database Connection"

Browser Click Continue



Task Configure the Standard Database Settings

Browser Select "MySQL" in the Database Driver presets.

Browser Edit the Database URL to "jdbc:mysql://localhost:3306/openfire"

Browser Edit the Username to "openfire"

Browser Edit the Password to "openfire"

Browser Click Continue

Note The database settings reflect those configured earlier.

The screenshot shows a web browser window with the URL `http://localhost:9090/setup/setup-datasource-standard.jsp`. The browser's address bar and tabs are visible. The Openfire 3.6.4 logo is in the top right corner. The main content area is titled "Database Settings - Standard Connection". It includes a "Setup Progress" sidebar on the left with a green bar indicating progress. The main form contains the following fields:

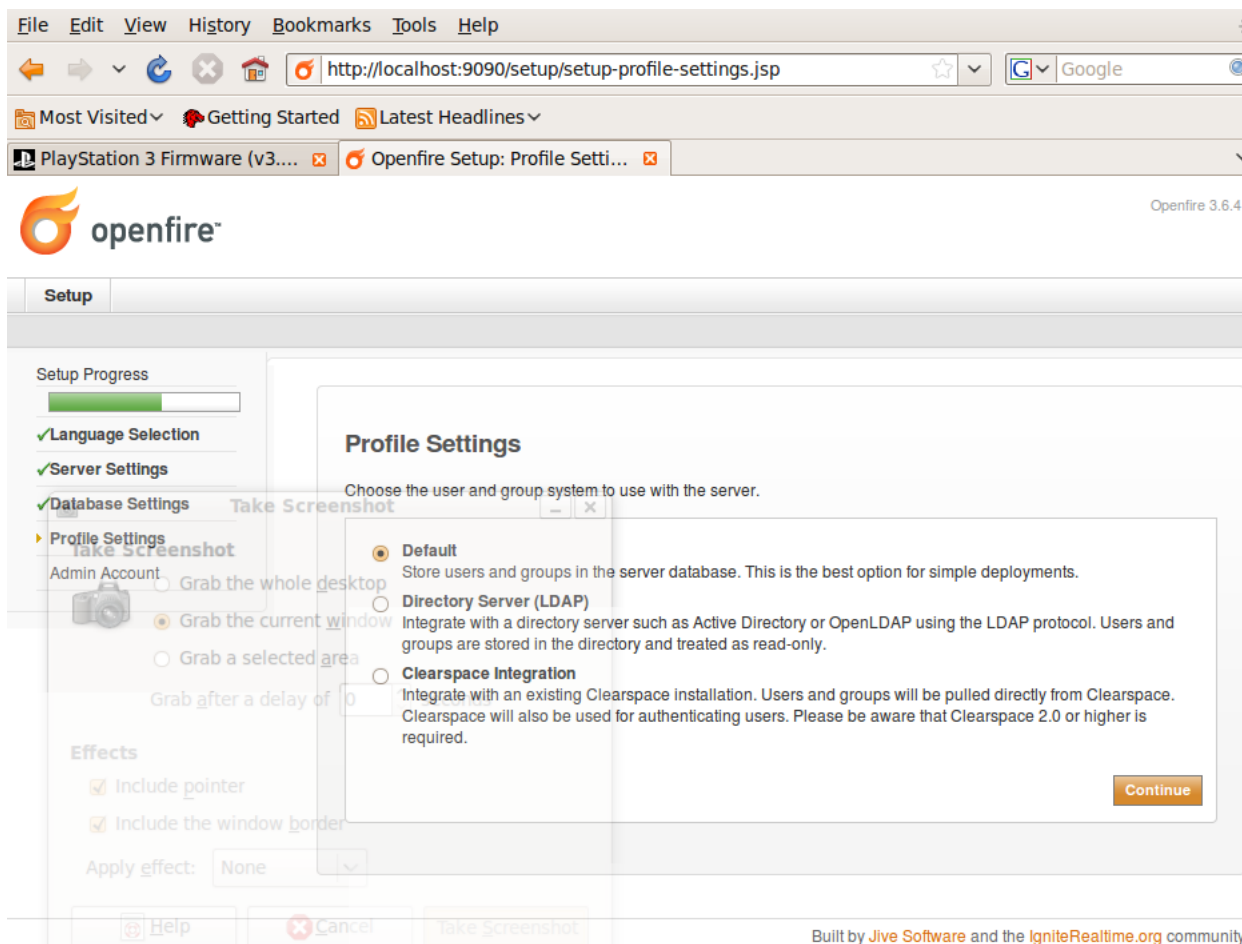
- Database Driver Presets:
- JDBC Driver Class:
- Database URL:
- Username:
- Password:
- Minimum Connections:
- Maximum Connections:
- Connection Timeout: Days

A note at the bottom states: "Note, it might take between 30-60 seconds to connect to your database." A "Continue" button is located at the bottom right of the form.

Task Configure the Profile Settings

Browser Select "Default"

Browser Click Continue



Task Configure the Administrator Settings

Browser Click "Skip this step"

Note This is only to complete the installation configuration. In the next steps a database import with change this value to the default.

File Edit View History Bookmarks Tools Help

http://localhost:9090/setup/setup-admin-settings.jsp

Most Visited Getting Started Latest Headlines

PlayStation 3 Firmware (v3.... Openfire Setup: Administrat...

openfire™ Openfire 3.6.4

Setup

Setup Progress

Language Selection

Server Settings

Database Settings

Profile Settings

Admin Account

Take Screenshot

Grab the whole desktop

Grab the current window

Grab a selected area

Grab after a delay of 0 seconds

Effects

Include pointer

Include the window border

Apply effect: None

Administrator Account

Enter settings for the system administrator account (username of "admin") below. It is important to choose a password for the account that cannot be easily guessed -- for example, at least six characters long and containing a mix of letters and numbers. You can skip this step if you have already setup your admin account (not for first time users).

Admin Email Address: admin@example.com

A valid email address for the admin account.

New Password:

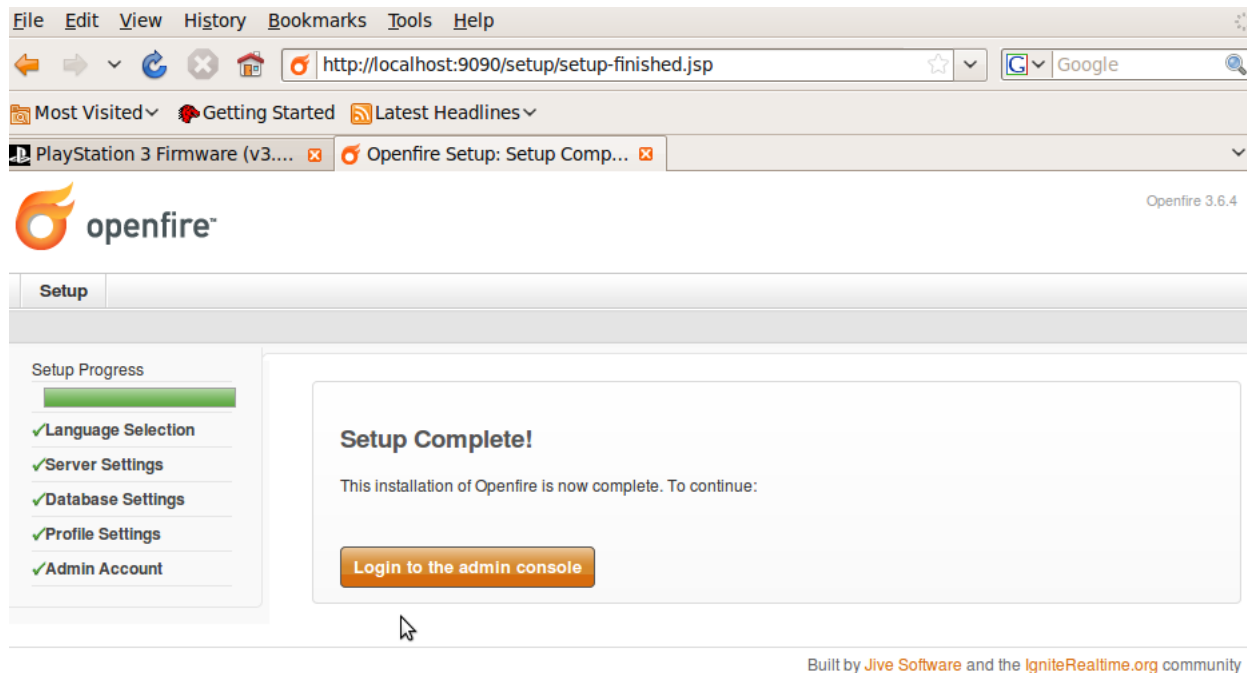
Confirm Password:

Skip This Step Continue

Built by Jive Software and the IgniteRealtime.org community

Task Setup Complete

Browser Setup should now be complete.



1.14 OpenFire Demo Configuration

The OpenFire server must be configured to match the requirements of the pre-canned demo applications. As such the existing database will be cleared and a valid configuration loaded in its place. This saves the long and tedious configurations required within OpenFire for all the users, groups and chatrooms.

Task Shutdown the OpenFire Server

Note Check the Openfire documentation to shutdown the server for your OS.

Task Clear the old database.

```
$ mysql -u root -p
```

```
mysql$ drop database openfire;
```

```
mysql$ create database openfire character set utf8;
```

```
mysql$ commit; exit;
```

Task Load the new database

```
$ mysql -u openfire -p openfire < $TINOS_HOME/documentation/demo/simple-ping/db/openfire.db
```

Note You will be prompted for the password : "openfire"

Task Start the OpenFire Server

Note Check the Openfire documentation to start the server for your OS.

Task Login to the OpenFire Administration Console

Browser Goto URL: http://localhost:9090

Browser Username : "admin", Password : "12345"

Note You can change the password afterwards if you wish.

Openfire 3.6.4
Logged in as admin - [Logout](#)

Server Manager | **Server Settings** | **Media Services**

Server Information

Below you will find server information, ports being used and latest news about Openfire.

Server Properties

- Server Uptime: Less than 1 minute -- started Sep 1, 2009 2:11:11 PM
- Version: Openfire 3.6.4
- Server Directory: /usr/share/openfire
- Server Name: localhost

Environment

- Java Version: 1.6.0_14 Sun Microsystems Inc. -- Java HotSpot(TM) 64-Bit Server VM
- Appserver: jetty-6.1.x
- Host Name: chimera
- OS / Hardware: Linux / amd64
- Locale / Timezone: en / Greenwich Mean Time (0 GMT)
- Java Memory: 65.68 MB of 442.69 MB (14.8%) used

Ignite Realtime News

- [Tinder 1.1.0 has been released](#), Aug 27, 2009
- [New Openfire monitoring plugin](#), Aug 18, 2009
- [Introducing Tinder, an XMPP object implementation library](#), Jun 23, 2009
- [Openfire 3.6.4 has been released](#), May 2, 2009
- [Openfire 3.6.3 has been released](#), Jan 9, 2009
- [Openfire 3.6.2 has been released](#), Nov 22, 2008

Server Ports

Interface	Port	Type	Description
All addresses	5222	Client to Server	The standard port for clients to connect to the server. Connections may or may not be encrypted. You can update the security settings for this port.
All addresses	5223	Client to Server	The port used for clients to connect to the server using the old SSL method. The old SSL method is not an XMPP standard method and will be deprecated in the future. You can update the security settings for this port.
All addresses	9090	Admin Console	The port used for unsecured Admin Console access.
All addresses	9091	Admin Console	The port used for secured Admin Console access.
All addresses	7777	File Transfer Proxy	The port used for the proxy service that allows file transfers to occur between two entities on the XMPP network.
All addresses	7070	HTTP Binding	The port used for unsecured HTTP client connections.
All addresses	7443	HTTP Binding	The port used for secured HTTP client connections.
	3478		

Done

1.15 Virgo Server Configuration

Task Configure logging/trace for the Jnode applications

```
$ . $HOME/env.sh
```

```
$ cd $SERVER_HOME
```

```
$ vi config/serviceability.xml
```

Note Add the following lines between the TINOS Start / End to the end of the serviceability.xml file. To exit "vi" use the key sequence [ESC]:wq[RETURN]

```
<!-- TINOS - Start -->
    <logger level="DEBUG" additivity="false" name="org.jnode">
        <appender-ref ref="SIFTED_LOG_FILE" />
    </logger>
    <logger level="DEBUG" additivity="false" name="org.pouzinsociety">
        <appender-ref ref="SIFTED_LOG_FILE" />
    </logger>
<!-- TINOS - End -->

    <root level="WARN">
        <appender-ref ref="SIFTED_LOG_FILE" />
```

Task Add the demo bundles

```
$ cp $TINOS_HOME/documentation/demo/simple-ping/bundles/* repository/usr
```

Task Add the Plan files

```
$ cp $TINOS_HOME/documentation/demo/simple-ping/plans/* .
```

1.16 XMPP/Jabber IM Client

Once configured you can use the IM Client to visually see the presence of the nodes within the demo networks, as well as sit in the networks (via chatrooms) and see the interactions. Super cheap and cheerful GUI.

The following clients have been tested successfully (Ubuntu : Pidgin, OSX : iChat, Windows : Spark). However as long as your client supports XMPP/Jabber, it should be fine.

Task Configure Your IM Client (XMPP/Jabber Capable)

Account Details

Server Protocol: XMPP, Domain: localhost, Server: localhost

Buddy User: human, Password: Human

Note Below are sample configuration screens from the Pidgin client.

The image displays two side-by-side screenshots of the Pidgin client's configuration windows. Both windows have a 'Basic' tab selected.

Left Window (Basic Tab):

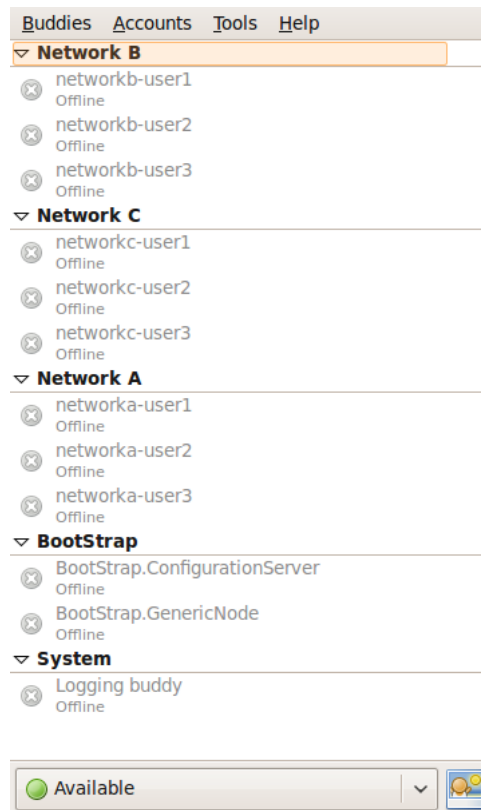
- Login Options:**
 - Protocol: XMPP (dropdown)
 - Username: human (text box)
 - Domain: localhost (text box)
 - Resource: (empty text box)
 - Password: (password field with 6 dots)
 - ☒ Remember password
- User Options:**
 - Local alias: (empty text box)
 - ☐ New mail notifications
 - ☐ Use this buddy icon for this account:
 - Buddy icon: (small square icon)
 - Remove button: (button with minus sign)
- ☐ Create this new account on the server
- Buttons: Cancel (with red X), Add (with plus sign)

Right Window (Advanced Tab):

- XMPP Options:**
 - ☐ Require SSL/TLS
 - ☐ Force old (port 5223) SSL
 - ☐ Allow plaintext auth over unencrypted streams
 - Connect port: 5222 (text box)
 - Connect server: localhost (text box)
 - File transfer proxies: proxy.jabber.org (text box)
 - ☒ Show Custom Smileys
- Proxy Options:**
 - Proxy type: Use GNOME Proxy Settings (dropdown)
- ☐ Create this new account on the server
- Buttons: Cancel (with red X), Add (with plus sign)

Task Ensure you can see Offline Buddies

Note Sample Buddy Roster from Pidgin Client.



1.17 TINOS - Simple Ping Demo

For the first initial loading of the demo - the following actions must be performed in exact sequence. This is primarily to make life easier for the person giving / viewing the demo as everything will start in an order that will match a presentation.

Task Start the Virgo Server

```
$ . $HOME/env.sh
```

```
$ cd $SERVER_HOME/bin
```

```
$ ./startup.sh -clean
```

Note Wait for this to complete.

If you are doing a demo - it is most useful to have the IM Client open during the demo as you will be able to watch the bootstrap in progress (via presence) and also the nodes as they are configured and come online.

The demo scenario is the almost the most basic possible with a simple ping scenario being enacted between the nodes but in order to do this the nodes, drivers, stacks (IPv4/TCP/Socket API/Name Resolver/Routes) and simple traffic generator (ping) are configured and enabled within the OSGi environment.

This is a starting point for more elaborate scenarios but more importantly it provides a simple demo that validates environment setup is correct.

1.17.1 Initial Run of the Demo

Task Start Your IM Client & login as human

Task New Terminal Shell - Load the Demo Applications

```
$ . $HOME/env.sh
```

```
$ cd $SERVER_HOME
```

Task Start the BootStrap configuration Manager

```
$ mv org.pouzinsociety.config.manager.plan pickup
```

Note Watching the other terminal - wait until the application in successfully loaded.

Task Start the Logger Agent

```
$ mv org.pouzinsociety.logger.plan pickup
```

Note Watching the other terminal - wait until the application in successfully loaded.

Task Start Jnode0

```
$ mv org.pouzinsociety.node.plan pickup
```

Note Watching the other terminal - wait until the application in successfully loaded.

Note Wait for IM Buddy - networkb-user1 to come online (Jnode0 fully configured).

Task Start Jnode1

```
$ mv org.pouzinsociety.node1.plan pickup
```

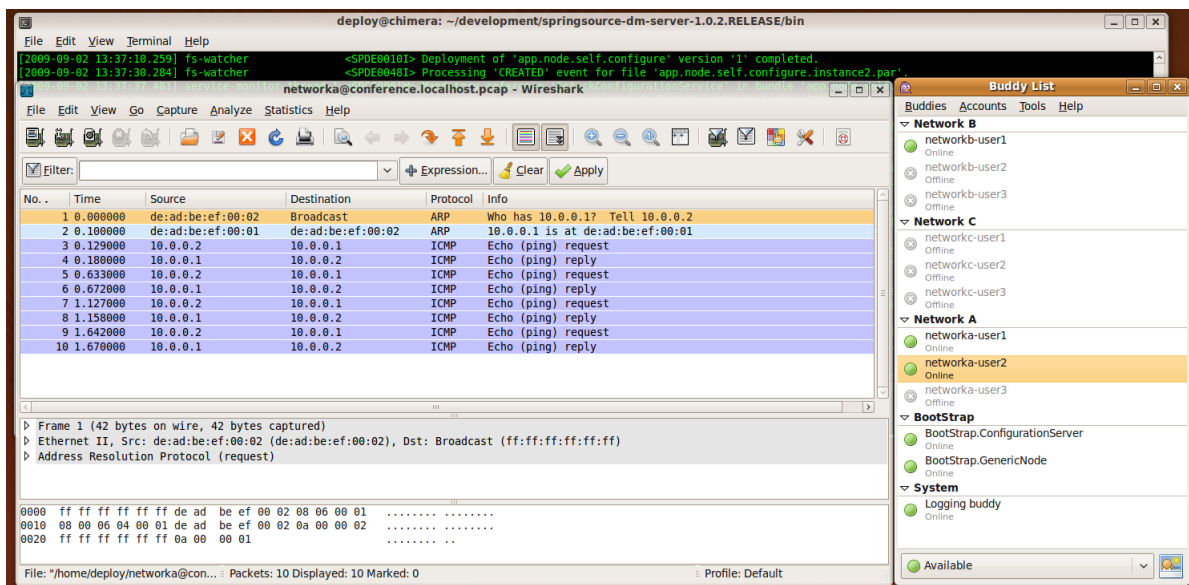
Note Watching the other terminal - wait until the application in successfully loaded.

Note Wait for IM Buddy - networkb-user2 to come online (Jnode1 fully configured).

Note The logger application will produce PCAP files in the "/tmp" directory.

Task Start WireShark

Note The file to load into WireShark is "/tmp/networka@conference.localhost.pcap" as this will have the ARP/Ping traffic present in it.



1.17.2 Stopping the Virgo Server

Task Stop the Virgo Server

Note Simply CTRL-C in the shell you started the server.

1.17.3 Post-Initial Demo

Task Running it Again

Note Simply start the server. Do not copy demo plan files into the pickup directory. The server will automatically pick them up on all the subsequent server startups.

Note Delete the pcap files under the "/tmp" directory.

1.18 Useful Links

Note Free Book - OSGi in Practise

URL <http://neilbartlett.name/blog/osgibook/>

—

Note SpringSource Enterprise Bundle Repository

URL <http://www.springsource.com/repository/app/>

–

Note SpringSource dm Server Programmer Guide

URL <http://static.springsource.com/projects/dm-server/1.0.x/programmer-guide/htmlsingle/programmer-guide.html>

–

Note OSGi Service Platform R4 Specification

URL <http://www.osgi.org/Download/Release4V41?info=nothanks>

–

Note Tutorial for Spring Dynamic Modules (DM) for OSGi Service Platforms

URL <http://springosgi.googlepages.com/>