**MMUCU Management System**

**FasTech Solution**

Contents

[1. System Requirements 2](#_Toc432611154)

[2. Installation and configuration 2](#_Toc432611155)

[ **Install Java** 2](#_Toc432611156)

[ **Install Wildfly** 5](#_Toc432611157)

[ **Install Jar files** 5](#_Toc432611158)

[ **Install Postgres** 6](#_Toc432611159)

[ **Install ant** 8](#_Toc432611160)

[ **Install Subversion** 9](#_Toc432611161)

[ **Populate the database** 9](#_Toc432611162)

[4. Application Deployment 9](#_Toc432611163)

**Guide**

# System Requirements

Operating System

* Debian or derivative e.g. Ubuntu, Mint
* Windows 7 and above

Deployment Applications

* WildFly 8 and above
* JDK 1.8 and above
* Java Ant
* PostgreSQL 9 and above
* Client Subversion Tools - "subversion-tools" packages in Linux or TortoiseSVN in Windows

Developer Applications

* Eclipse JavaEE IDE
* Jar file collection
* Text Editor

# Installation and configuration

# **Install Java**

This tutorial will cover the installation of 32-bit and 64-bit Oracle Java 7 (currently version number 1.8.0\_25) JDK on 32-bit and 64-bit Ubuntu operating systems. These instructions will also work on Debian and Linux Mint.

1. **Check to see if your Ubuntu Linux operating system architecture is 32-bit or 64-bit, open up a terminal and run the following command below.**

***Type/Copy/Paste:*** *file /sbin/init*

1. **Check if you have Java installed on your system**. To do this, you will have to run the Java version command from terminal.

***Type/Copy/Paste:*** *java –version*

* **If you have OpenJDK installed on your system it may look like this:**

*Java version "1.7.0\_15"  
OpenJDK Runtime Environment (IcedTea6 1.10pre) (7b15~pre1-0lucid1)  
OpenJDK 64-Bit Server VM (build 19.0-b09, mixed mode)*

1. **Completely remove the OpenJDK/JRE from your system and create a directory to hold your Oracle Java JDK/JRE binaries.** This will prevent system conflicts and confusion between different vendor versions of Java. For example, if you have the OpenJDK/JRE installed on your system, you can remove it by typing the following at the command line and then create a path for the next installation:

*sudo apt-get purge openjdk-\\**

*sudo mkdir -p /opt/Programs/jdks/*

1. [**Download the Oracle Java JDK/JRE for Linux**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)**.** Make sure you select the **correct** compressed binaries for your system architecture 32-bit or 64-bit (which end in tar.gz).

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

1. **Copy the Oracle Java binaries into the /opt/Programs/jdks directory.** In most cases, the Oracle Java binaries are downloaded to: **/home/"your\_user\_name"/Downloads.**

* **32-bit Oracle Java on 32-bit Ubuntu Linux installation instructions: Type the following in a terminal**

***Type/Copy/Paste:*** *cd /Downloads*

***Type/Copy/Paste:*** *sudo cp -r jdk-8u40-linux-i586.tar.gz /opt/Programs/jdks*

***Type/Copy/Paste:*** *cd /opt/Programs/jdks*

1. **64-bit Oracle Java on 64-bit Ubuntu Linux installation instructions:**

***Type/Copy/Paste:*** *cd ~/Downloads*

***Type/Copy/Paste:*** *sudo cp -r jdk-8u40-linux-x64.tar.gz /opt/Programs/jdks*

***Type/Copy/Paste:*** *cd /opt/Programs/jdks*

1. **Unpack the compressed Java binaries, in the directory /opt/Programs/jdks**

***Type/Copy/Paste:*** *sudo tar xvzf jdk-8u40-linux-i586.tar.gz*

***Or***

***Type/Copy/Paste****: sudo tar xvzf jdk-8u40-linux-x64.tar.gz*

**Double-check your directories.** At this point, you should have one uncompressed binary directory in /opt/Programs/jdks for the Java JDK listed as:

***Type/Copy/Paste:*** *ls –al /opt/Programs/jdks*

***jdk1.8.0\_25***

1. **Edit the system PATH file /etc/profile and add the following system variables to your system path**. Use nano, gedit or any other text editor, as root, open up /etc/profile.

***Type/Copy/Paste:*** *sudo gedit /etc/profile*

*Or*

***Type/Copy/Paste:*** *sudo nano /etc/profile or sudo gedit ./bashrc*

1. **Scroll down to the end of the file using your arrow keys and add the following lines below to the end of your /etc/profile file:**

***Type/Copy/Paste:***

*JAVA\_HOME=/opt/Programs/jdks/jdk1.8.0\_40*

*JRE\_HOME=$JAVA\_HOME/jre*

*PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin*

*JBOSS\_HOME=/opt/Programs/WildFly/8.2.0*

*export JBOSS\_HOME*

*export JAVA\_HOME*

*export JRE\_HOME*

*export PATH*

1. **Inform your Ubuntu Linux system where your Oracle Java JDK/JRE is located.** This will tell the system that the new Oracle Java version is available for use.

**Type/Copy/Paste:** *sudo update-alternatives --install "/usr/bin/java" "java" "/opt/Programs/jdks/jdk1.8.0\_40/jre/bin/java" 1*

1. **Inform your Ubuntu Linux system that Oracle Java JDK/JRE must be the default Java.**

***Type/Copy/Paste:*** *sudo update-alternatives --set java /opt/Programs/jdks/jdk1.8.0\_40/jre/bin/java*

1. **Reload your system wide PATH /etc/profile by typing the following command:**

***Type/Copy/Paste****: . /etc/profile*

Note your system-wide PATH /etc/profile file will reload after reboot of your Ubuntu Linux system

# **Install Wildfly**

1. Create a path where you will install your JBoss Wildfly application server

***Type/Copy/Paste****: sudo mkdir /opt/Programs/WildFly/*

1. Download JBoss Wildfly and extract the zip file to the above directory.

<http://download.jboss.org/wildfly/8.2.0.Final/wildfly-8.2.0.Final.zip>

1. Copy the directory to /opt/Programs/WildFly

*cp wildfly-8.2.0.Final.zip /opt/Programs/WildFly*

*unzip wildfly-8.2.0.Final.zip*

*mv wildfly-8.2.0.Final 8.2.0*

1. Start Wildfly

*cd /opt/Programs/WildFly/8.2.0/bin*

*./standalone.sh*

1. Add user

*cd /opt/Programs/WildFly/8.2.0/bin*

*./add-user.sh*

# **Install Jar files**

1. Create a directory /opt/Jars

*mkdir /opt/Jars*

1. Download the jar files and extract the zip

*cd /opt/Jars*

# **Install Postgres**

Postgresql is the database Server. To install

**As root:**

1. Add the PostgreSQL Apt Repository create a Debian sources list file (e.g. pgdg.list)

*# Install Pgadmin3*

*#Add the PGDB APT Source file From the Terminal:*

*sudo touch /etc/apt/sources.list.d/pgdg.list*

*#Open the pgdg.list File in gedit (use sudo):*

*sudo gedit /etc/apt/sources.list.d/pgdg.list*

*#then add the following line in gedit and save (where I used “trusty” below,*

*#use the name of your release for Ubuntu, or the corresponding Ubuntu release*

*#if you you are using Linux Mint):*

*deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main*

*#Save, and close gedit.*

*#Next, add the package repository key:*

*sudo apt-get install wget ca-certificates*

*wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -*

*#Update Package Sources:*

*sudo apt-get update*

*#Install Postgres:*

*sudo apt-get install postgresql-9.4 pgadmin3*

1. **Add a user and database in Postgres**To create a normal user and an associated database you need to follow the procedure below:

**Step #1: Becoming a superuser**

You need to login as database super user under postgresql server. Again the simplest way to connect as the postgres user is to change to the postgres unix user on the database server using su command as follows:

*# su*

*# su - postgres*

**Step #2: Now connect to database server**

**Step #3:**

*#####################*

*Create database*

*#####################*

*1) psql template1*

*CREATE USER cu WITH PASSWORD 'Cu12c';*

*ALTER ROLE cu WITH CREATEDB;*

*\q*

*psql template1*

*CREATE DATABASE cudb;*

*GRANT ALL PRIVILEGES ON DATABASE cudb to cu;*

*\q*

*psql template1*

*ALTER DATABASE cudb OWNER to cu;*

# **Install ant**

Retrieve the binary distribution of Java Ant from the Apache website: [http://ant.apache.org](http://ant.apache.org/)

Before installing Ant, ensure that you have installed Java and that the environment variable ‘JAVA\_HOME’ resolves correctly.

Assuming you downloaded Ant version 1.9.6:

sudo gedit ~/.bashrc

JAVA\_HOME=/opt/Programs/jdks/jdk1.8.0\_60

export PATH=${JAVA\_HOME}/bin:$PATH

export JBOSS\_HOME=/opt/Programs/WildFly/8.2.0

ANT\_HOME="/opt/Programs/apache-ant-1.9.4"

PATH=$PATH:$HOME/bin:$ANT\_HOME/bin

export ANT\_HOME PATH

With ant -version you should get an output like

Apache Ant(TM) version 1.9.6 compiled on June 29 2015

# **Install Subversion**

1. To install subversion:

*sudo apt-get install subversion*

1. Create a directory on /home called svn

*mkdir ~/svn*

*cd svn/*

1. Checkout *School* project

*svn checkout https://github.com/msomi22/ChristianUnion*

# **Populate the database**

1. Go to the School project on subversion folder

*Populate the database*

*cd ~/svn/ChristianUnion/trunk/webapp/bin*

*###########Execute the script to populate the database*

*./dbSetup.sh*

# Application Deployment

To deploy the  *ChristianUnion*

*cd ~/svn/**ChristianUnion/trunkwebapp/*

*ant dist*

Check the logs as the project is deployed. If it is successful, you can access the *ChristianUnion* system from your browser by accessing the following URL: