

VIVO 1.1 Installation Instructions - Debian

Installation instructions for VIVO 1.1 on Debian OS.

Obsolete: This page was written for VIVO 1.1 - it is no longer correct.



We will attempt to revise this page to work with newer releases. In the meantime, please check the current installation instructions: [VIVO Installation Instructions](#)

Install Apache 2 and Tomcat 6

Add 'non-free' to end of deb and deb-src

```
nano /etc/apt/sources.list
```

Update and install java

```
apt-get update apt-get install apache2 sun-java6-jdk sun-java6-jre libtcnative-1 sudo
```

Ensure that Java was installed

```
java -version
```

Set JAVA_HOME

```
nano ~/.bashrc
```

Add the following at the end of the file, making sure to check the version

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun export CLASSPATH=$CLASSPATH:/usr/share/java/mysql-connector-java-5.1.6.jar export CLASSPATH=$CLASSPATH:/usr/share/java/mysql-connector-java.jar
```

Install Tomcat 6 Note: get the cores tar.gz

```
wget http://people.apache.org/~remm/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.tar.gz tar -zxvf apache-tomcat-6.0.20.tar.gz mv apache-tomcat-6.0.20/ /usr/local/tomcat
```

Create tomcat Group/user

```
groupadd tomcat useradd -g tomcat -d /usr/local/tomcat tomcat usermod -G www-data tomcat
```

Set Password

```
passwd tomcat VIVO123
```

Create the INIT File for Tomcat with the following text added:

```
nano /etc/init.d/tomcat # Tomcat auto-start description: Auto-starts tomcat processname: tomcat pidfile: /var/run/tomcat.pid export JAVA_HOME=/usr/lib/jvm/java-6-sun case $1 in start) sh /usr/local/tomcat/bin/startup.sh ;; stop) sh /usr/local/tomcat/bin/shutdown.sh ;; restart) sh /usr/local/tomcat/bin/shutdown.sh sh /usr/local/tomcat/bin/startup.sh ;; esac exit 0
```

Adjust permissions

```
chmod 755 /etc/init.d/tomcat
```

Add for auto-start on boot

```
update-rc.d tomcat defaults
```

Modify the default users file with the following:

```
nano /usr/local/tomcat/conf/tomcat-users.xml <tomcat-users> <role rolename="manager"/> <role rolename="admin"/>
<user username="VIVO" password="VIVO123" roles="admin,manager"/> </tomcat-users>
```

Restart tomcat

```
/etc/init.d/apache2 stop /etc/init.d/tomcat restart /etc/iinit.d/apache2 start
```

Test Admin

```
http://localhost:8080/manager/html/list
```

Enter username/password as shown here

```
Username: VIVO Password: VIVO123
```

Install VIVO

Create a VIVO user for deployment

```
useradd vitro passwd vitro
```

Enter new password:

```
vitrol23
```

Add user to sudoers file

```
nano /etc/sudoers
```

Change the following line

```
root,vitro ALL=(ALL) ALL
```

Stop Tomcat

```
/etc/init.d/tomcat stop
```

Install more required packages

```
apt-get update apt-get install subversion-tools sudo ant mysql-server-5.0 libmysql-java
```

Mysql root password is set to

```
vitrol23
```

Download latest repository of VIVO

```
cd /tmp wget https://downloads.sourceforge.net/project/vivo/VIVO%20Application%20Source/rel-1.1.tar.gz tar -
zxvf rel-1.1.tar.gz
```

Move downloaded directory

```
mv vivo-rel-1.1 /usr/local/vivo
```

Edit deploy.properties

```
cd /usr/local/vivo mv example.deploy.properties deploy.properties nano deploy.properties VIVO.
defaultNamespace=http://localhost/vivo/ VIVOConnection.DataSource.url=jdbc:mysql://localhost/VIVODB
VIVOConnection.DataSource.username=VIVO VIVOConnection.DataSource.password=VIVO123
```

Create directories as required in deploy.properties

```
mkdir /usr/local/vivo/data mkdir /usr/local/vivo/data/uploads mkdir /usr/local/vivo/data/luceneIndex chown
tomcat:www-data /usr/local/vivo/data/uploads chmod 775 /usr/local/vivo/data/uploads chown tomcat:www-data /usr
/local/vivo/data/luceneIndex chmod 775 /usr/local/vivo/data/luceneIndex
```

Edit globalbuild.properties in config

```
cd /usr/local/vivo/vitro-core/config mv example.globalbuild.properties globalbuild.properties nano globalbuild.
properties java_api=/usr/lib/jvm/java-6-sun
```

Create the source.home directory as required in globalbuild.properties file

```
mkdir /usr/local/src/Vitro chown tomcat:www-data /usr/local/src/Vitro chmod 775 /usr/local/src/Vitro
```

Create an empty MySQL database (password vitro123)

```
mysql -u root -p CREATE DATABASE vitrodB CHARACTER SET utf8; CREATE USER 'VIVO'@'localhost' IDENTIFIED BY
'vitro123'; GRANT ALL ON *.* TO 'VIVO'@'localhost'; quit;
```

Deploy vivo and start tomcat

```
/etc/init.d/tomcat stop cd /usr/local/vivo ant all
```

After a successful build, start Tomcat

```
/etc/init.d/tomcat start
```

Test application at

```
http://localhost:8080/vivo
```

Installation of Mod_Proxy

Enable the following Modules in Apache

```
a2enmod proxy a2enmod proxy_http
```

Setup support in your servlet container

Note: Be careful that there is no direct HTTP listener opened by the servlet container. If, for example, there's an HTTP connector listening on port 8080 and no interceding firewall, users would be able to directly access the servlet on port 8080, which bypasses Apache. This also means they would bypass Shibboleth authentication and authorization.

```
nano /usr/local/tomcat/conf/server.xml
```

Make sure the following is in your server.xml file

```
<!-- Set context path for access to 8080 over port 80 --> <Connector port="8080" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75" enableLookups="false" redirectPort="8443"
acceptCount="100" connectionTimeout="10000" disableUploadTimeout="true" proxyName="YOUR.DOMAIN.EDU" proxyPort="
80"/> <!-- Set context path for access over SSL --> <Connector port="8443" maxHttpHeaderSize="8192" maxThreads="
150" minSpareThreads="25" maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true" acceptCount="
100" scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" proxyName="YOUR.DOMAIN.EDU" proxyPort="
443" /> <!-- Set docBase for root webserver application --> <Host name="localhost" appBase="webapps/vivo">
<Context path="" docBase="." /> </Host>
```

Configure Apache to route requests to your servlet

Modify your default Apache site configuration to map requests on the proper virtual hosts to your application.

```
nano /etc/apache2/sites-available/default ProxyRequests On ProxyPreserveHost On ProxyStatus On <Proxy *> Order
deny,allow Allow from all </Proxy> # Use proxy to restrict the need for a port in the URL ProxyPass /vivo/
http://localhost:8080/vivo/ ProxyPassReverse /vivo/ http://localhost:8080/vivo/ ProxyPass / http://localhost
/vivo/ ProxyPassReverse / http://localhost/vivo/
```

Now repeat if your virtual host will be SSL enabled.

```
nano /etc/apache2/sites-available/default-ssl
```

Add the following to your virtual host

```
ProxyRequests On ProxyPreserveHost On ProxyStatus On <Proxy *> Order deny,allow Allow from all </Proxy> # Use
proxy to restrict the need for a port in the URL ProxyPass /vivo/ http://localhost:8080/vivo/ ProxyPassReverse
/vivo/ http://localhost:8080/vivo/ ProxyPass / http://localhost/vivo/ ProxyPassReverse / http://localhost/vivo/
```

// Stop and Restart Apache/Tomcat

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

Installing Shibboleth 2

Install the Shibboleth 2.0 Service Provider Packages

```
apt-get update apt-get install shibboleth-sp2-schemas libshibsp-dev apt-get install libshibsp-doc libapache2-
mod-shib2 opensaml2-tools
```

Enter Shibboleth config directory

```
cd /usr/sbin/
```

Generate a key/certificate for Shibboleth

```
./shib-keygen -h shib.your.domain.edu
```

Make an SSL directory to store the certs

```
mkdir /etc/shibboleth/ssl
```

Copy certs and rename with your hostname

```
cp -rp /etc/shibboleth/sp-cert.pem /etc/shibboleth/ssl/YOUR.DOMAIN.EDU.cert cp -rp /etc/shibboleth/sp-key.pem /etc/shibboleth/ssl/YOUR.DOMAIN.EDU.pem
```

Rename the default XML file

```
cd /etc/shibboleth mv shibboleth2.xml shibboleth2.xml.bak
```

Download the Linux XML config file from your Identity Provider (IDP) at your institution

```
wget http://YOUR.IDENTITYPROVIDER.EDU/linux.shibboleth2.xml
```

Rename the XML config file

```
mv linux.shibboleth2.xml shibboleth2.xml
```

Configure your XML file per your organizations Identity Provider

You will need to obtain a URN from your IDP.

Enabling Shibboleth Authentication

Add a line to your Apache configuration on the proper virtual host, such as in httpd.conf, to trigger Shibboleth session initiation and authentication for your application. The use of ShibUseHeaders On is important.

Edit virtual host

```
nano /etc/apache2/sites-available/default
```

Add the following to your virtual host.

You can enter anything to replace "shibauth". For example, you could use "/secure" or just "/" to secure the entire virtual host.

```
# Path for authentication <Location /shibauth> AuthType shibboleth ShibRequireSession On ShibUseHeaders On
require valid-user </Location> # Shib Variables Available to entire web server <Location /> AuthType shibboleth
ShibRequireSession Off require valid-user ShibUseHeaders On require shibboleth </Location>
```

Repeat for SSL enabled site, edit virtual host

```
nano /etc/apache2/sites-available/default-ssl
```

Add the following to your virtual host

```
<Location /shibauth> AuthType shibboleth ShibRequireSession On ShibUseHeaders On require valid-user </Location>
# Shib Variables Available to entire web server <Location /> AuthType shibboleth ShibRequireSession Off require
valid-user ShibUseHeaders On require shibboleth </Location>
```

Installing Shibboleth Authentication Plugin

Create the following files:

```
uf_login_process.jsp webapp/src/edu/cornell/mannlib/VIVO/webapp/controller/edit/UfAuthenticate.java (Copy of
Authenticate.java, Compiled)
```

Modify the following files:

```
about_body.jsp (Add html link to UF Shib Login) <strong>UF Shibboleth Login</strong><br/> <a href="
http://vivotest.ctrrip.ufl.edu/Shibboleth.sso/Login? target=<%= URLEncoder.encode("http://vivotest.ctrrip.ufl.edu:
8080/VIVO/uf_login_process.jsp" target="_self">Login &raquo;</a> webapp/WEB-INF/web.xml (Modified to reflect
new class) <servlet> <servlet-name>ufauthenticate</servlet-name> <servlet-class>edu.cornell.mannlib.VIVO.webapp.
controller.edit.UfAuthenticate</servlet-class> </servlet> <servlet-mapping> <servlet-name>ufauthenticate<
/servlet-name> <url-pattern>/ufauthenticate</url-pattern> </servlet-mapping>
```