A simple installation

How to get VIVO up and running on your computer, for testing or experimentation, or just to learn how to do it.

If you want to install VIVO on a production server, or if you want to develop VIVO code, you should also read the section on Installation options.

Preparing for VIVO

Install required software

Before installing VIVO, make sure that the following software is installed on your computer:

- Java (SE) 1.7.x http://java.sun.com
  - VIVO has not been tested with OpenJDK
- Apache Tomcat 6.x or 7.x http://tomcat.apache.org
- Apache Ant 1.8 or higher, http://ant.apache.org
- MySQL 5.1 or higher, http://www.mysql.com

Set up the environment variables for `JAVA_HOME` and `ANT_HOME` and add the executables to your path, as required. This requirement depends on the operating system you are using. Consult the installation directions from the software support websites.

The following browsers are supported for this release

- Mac:
  - Chrome 30.0.1599.69 and above
  - FireFox 3.6.28, 10.0.12, 24
  - Opera 12.02
  - Safari 5.0.3
- PC:
  - Chrome 25.1364.2 and above
  - FireFox 10.0.12, 24
  - Internet Explorer 8, 9, 10
  - Opera 12.02

Did it work?

You can test the software installation by typing these commands:

```
java -version
mysql --version
ant -version
```

Each of these command should print a response that tells you what version is installed. If any of these commands prints an error message, or reports an unexpected version number, you should review your installation.

Create an empty database and a database account

Decide on a database name, username, and password.You will need these values for this step, and again when you Specify runtime properties.

Log into your MySQL server and create a new database in MySQL that uses `UTF-8` encoding. At the MySQL command line you can create the database and user with these commands substituting your values for `dbname`, `username`, and `password`. Most of the time, the hostname will be `localhost`.

```
CREATE DATABASE dbname CHARACTER SET utf8;
GRANT ALL ON dbname.* TO 'username'@'hostname' IDENTIFIED BY 'password';
```

Building VIVO

Download the VIVO source code

Download the VIVO application source as either `rel-1.6.zip` or `rel-1.6.gz` file and unpack it on your web server: http://vivoweb.org/download

Specify build properties

At the top level of the VIVO distribution directory, rename the file `example.build.properties` to `build.properties`. Edit the file to suit your installation, as described in the following section.

These properties are used in compiling VIVO and deploying it to Tomcat. They will be incorporated into VIVO when it is compiled. If you want to change these properties at a later date, you will need to stop Tomcat, repeat the Compile and deploy step, and restart Tomcat.

Windows: For those installing on a Windows operating system, include the windows drive, but use the forward slash "/" and not the back slash "\" in the directory locations, e.g. `c:/tomcat`.

Property name	**`vitro.core.dir`**
Description	The directory where Vitro code is located. In the simple installation, this is set to `./vitro-core`
Default value	NONE
Example value	`./vitro-core`

Property name	**`tomcat.home`**
Description	The directory where tomcat is installed.
Default value	NONE
Example value	`/usr/local/tomcat`

Property name	**`webapp.name`**
Description	The name of your VIVO application. This is not a name that will be displayed to the user. This name appears in the URL used to access VIVO, and in the path to the VIVO directory within Tomcat.
Default value	NONE
Example value	`vivo`

Property name	**`vitro.home`**
Description	The directory where VIVO will store the data that it creates. This includes uploaded files (usually images) and the Solr search index. Be sure this directory exists and is writable by the Tomcat service.
Default value	NONE
Example value	`/usr/local/vivo/home`

Compile and deploy

In the previous step, you defined the location of the VIVO home directory, by specifying `vitro.home` in the `build.properties` file. If that directory does not exist, create it now.

At the command line, from the top level of the VIVO distribution directory, type:

```
ant all
```

to build VIVO and deploy to Tomcat's webapps directory.

The build script may run for as much as five minutes, and creates more than 100 lines of output. The process includes several steps:

- collecting the source files from the distribution directory,
- compiling the Java source code,
- running unit tests,
- preparing the Solr search engine,
- deploying VIVO and Solr to Tomcat.

Did it work?

If the output ends with a success message, the build was successful. Proceed to the next step.

```
BUILD SUCCESSFUL

Total time: 1 minute 49 seconds
```

If the output ends with a failure message, the build has failed. Find the cause of the failure, fix the problem, and run the script again.

```
BUILD FAILED

Total time: 35 seconds
```

The output of the build may include warning messages. The Java compiler may warn of code that is outdated. Unit tests may produce warning messages, and some tests may be ignored if they do not produce consistent results. If the output ends with a success message, these warnings may be ignored.

What user account owns the VIVO directories?

In many operating systems, the issue of file permissions is important. Who owns the files? Who is authorized to read them, or to write new files?

When running the VIVO build script, it must have permission to read and write to:

- the VIVO distribution directory
- the Tomcat webapps directory
- the VIVO home directory

When VIVO is started under Tomcat, Tomcat must have permission to read and write to:

- the Tomcat webapps directory
- the VIVO home directory

There are several ways to make this work. People who are experimenting with VIVO often use their own account to create the VIVO distribution directory, to run the build script, and to run Tomcat.

In more formal environments, it may be necessary to run Tomcat as a service, under its own account. In that case, some people choose to run the build script with `root` privilege, and then assign the resulting files to Tomcat:

```
sudo ant all
sudo chown -R tomcat /usr/local/vivo/home
sudo chown -R tomcat /usr/local/tomcat/webapps/vivo*
```

When installing on Microsoft Windows, this is not usually a problem.

Running VIVO

Configure Tomcat

Set JVM parameters

VIVO copies small sections of your RDF database into memory in order to serve Web requests quickly (the in-memory copy and the underlying database are kept in synch as edits are performed).

VIVO may require more memory than that allocated to Tomcat by default. With most installations of Tomcat, the `setenv.sh` or `setenv.bat` file in Tomcat's `bin` directory is a convenient place to set the memory parameters. *If this file does not exist in Tomcat's bin directory, you can create it.*
For example:

setenv.sh

```
export CATALINA_OPTS="-Xms512m -Xmx512m -XX:MaxPermSize=128m"
```

This tells Tomcat to allocate an initial heap of 512 megabytes, a maximum heap of 512 megabytes, and a PermGen space of 128 megs. Lower values may be sufficient, especially for small test installations.

If an `OutOfMemoryError` occurs during VIVO execution, increase the heap parameters and restart Tomcat.

Set security limits

VIVO is a multithreaded web application that may require more threads than are permitted under your Linux installation's default configuration. Ensure that your installation can support the required number of threads by making the following edits to `/etc/security/limits.conf`:

```
apache        hard        nproc        400
tomcat6        hard        nproc        1500
```

Set URI encoding

In order for VIVO to correctly handle international characters, you must configure Tomcat to conform to the URI standard by accepting percent-encoded UTF-8.

Edit Tomcat's `conf/server.xml` and add the following attribute to each of the Connector elements: URIEncoding="UTF-8".

```
<Server ...>
  <Service ...>
    <Connector ... URIEncoding="UTF-8"/>
      ...
    </Connector>
  </Service>
</Server>
```

Some versions of Tomcat already include this attribute as the default.

Take care when creating Context elements

Each of the webapps in the VIVO distribution (VIVO and Solr) includes a "context fragment" file, containing some of the deployment information for that webapp.

Tomcat allows you to override these context fragments by adding Context elements to `server.xml`. If you decide to do this, be sure that your new Context element includes the necessary deployment parameters from the overridden context fragment.

See the section entitled Running VIVO behind an Apache server for an example of overriding the VIVO context fragment.

Specify runtime properties

The build process in the Compile and deploy step created a file called `example.runtime.properties` in your VIVO home directory (specified by `vitro.home` in the `build.properties` file). Rename this file to `runtime.properties` and edit the file to suit your installation, as described below.

These properties are loaded when VIVO starts up. If you want to change these properties at a later date, you will need to restart Tomcat for them to take effect. You will not need to repeat the Compile and deploy step.

Windows: For those installing on Windows operating system, include the windows drive and use the forward slash "/" and not the back slash "\" in the directory locations, e.g. `c:/tomcat`.

Basic properties

These properties define some fundamental aspects of your VIVO installation. Most sites will need to modify all of these values.

Property name	`Vitro.defaultNamespace`
Description	The default RDF namespace for this installation. VIVO installations make their RDF resources available for harvest using linked data. Requests for RDF resource URIs redirect to HTML or RDF representations as specified by the client. To make this possible, VIVO's default namespace must have a certain structure and begin with the public web address of the VIVO installation. For example, if the web address of a VIVO installation is http://vivo.example.edu/ the default namespace must be set to "http://vivo.example.edu/individual/" in order to support linked data. Similarly, if VIVO is installed at http://www.example.edu/vivo the default namespace must be set to "http://www.example.edu/vivo/individual/" *** The namespace must end with "individual/" (including the trailing slash).**
Default value	NONE
Example value	`http://vivo.mydomain.edu/individual/`

Property name	`rootUser.emailAddress`
Description	Specify the email address of the root user account for the VIVO application. This user will have an initial temporary password of `rootPassword`. You will be prompted to create a new password on first login. NOTE: The root user account has access to all data and all operations in VIVO. Data views may be surprising when logged in as the root user. It is best to create a Site Admin account to use for every day administrative tasks.
Default value	NONE
Example value	`vivoAdmin@my.domain.edu`

Property name	`VitroConnection.DataSource.url`
Description	Specify the JDBC URL of your database. Change the end of the URL to reflect your database name (if it is not "vitrodb").
Default value	NONE
Example value	`jdbc:mysql://localhost/vivo`

Property name	`VitroConnection.DataSource.username`
Description	Change the username to match the authorized user you created in MySQL.
Default value	NONE
Example value	`username`

Property name	`VitroConnection.DataSource.password`
Description	Change the password to match the password you created in MySQL.
Default value	NONE
Example value	`password`

Property name	`email.smtpHost`
Description	Specify an SMTP host that the application will use for sending e-mail (Optional). If this is left blank, the contact form will be hidden and disabled, and users will not be notified of changes to their accounts.
Default value	NONE
Example value	`smtp.servername.edu`

Property name	`email.replyTo`
Description	Specify an email address which will appear as the sender in e-mail notifications to users (Optional). If a user replies to the notification, this address will receive the reply. If a user's e-mail address is invalid, this address will receive the error notice. If this is left blank, users will not be notified of changes to their accounts.
Default value	NONE
Example value	`vivoAdmin@my.domain.edu`

Connecting to the Solr search index

VIVO and its search index are actually two distinct web applications, and the simple installation puts them both into the same instance of Tomcat. Even so, the VIVO webapp must be told how to reach the Solr webapp.

Property name	`vitro.local.solr.url`
Description	URL of Solr context used in local VIVO search. Should consist of: `scheme + servername + port + vivo_webapp_name + "solr"` In the standard installation, the Solr context will be on the same server as VIVO, and in the same Tomcat instance. The path will be the VIVO webapp.name (specified above) + "solr"
Default value	NONE
Example value	`http://localhost:8080/vivosolr`

Additional properties

The `runtime.properties` file can accept many additional properties, but they aren't necessary for this simple installation. If you choose any of the Install ation options, you will probably need to set some of those properties.

Start Tomcat

Most Tomcat installations can be started by running `startup.sh` or `startup.bat` in Tomcat's `bin` directory. Start Tomcat and direct your browser to `http://localhost:8080/vivo` to test the application. Note that Tomcat may require several minutes to start VIVO.

On start up VIVO will run some diagnostic tests. If a problem is detected the normal VIVO pages will redirect to a startup status page describing the problem. You can stop Tomcat, attempt to fix the problem and proceed from the Compile and deploy step. If the problem is not serious, the startup status page may offer a `continue` link which will allow you to use VIVO in spite of the problems.

If the startup was successful, you will see the VIVO home page.

If Tomcat does not start up, or the VIVO application is not visible, check the files in Tomcat's logs directory. Error messages are commonly found in *`[tomcat]`*`/logs/catalina.out`, *`[tomcat]`*`/logs/vivo.all.log` or *`[tomcat]`*`/logs/localhost.log`

Remember that Tomcat must have permission to read and write its own files, and the files in the VIVO home directory. This may mean that you must use a particular script or a particular user account to start Tomcat.

Start using VIVO

Log in and add RDF data

Direct your browser to the VIVO home page. Click the "Log in" link near the upper right corner. Log in with the `rootUser.emailAddress` that you set in the `runtime.properties` file. The initial password for the root account is `rootPassword`. When you first log in, VIVO will require you to change the password. When login is complete, the search index is checked and, if it is empty, a full index build will be triggered in the background, in order to ensure complete functionality throughout the site.

After logging in, you will be presented with a menu of editing options. Here you can create OWL classes, object properties, data properties, and configure the display of data. Currently, any classes you wish to make visible on your website must be part of a class group and any individual must have an rdfs: label. There are a number of visibility and display options available for classes and properites. VIVO comes with a core VIVO ontology, but you may also upload other ontologies from an RDF file.

Under the "Advanced Data Tools" click "Add/Remove RDF Data." Note that Vitro currently works best with OWL-DL ontologies and has only limited support for pure RDF data. You can enter a URL pointing to the RDF data you wish to load or upload from a file on your local machine. Ensure that the "add RDF" radio button is selected. You will also likely want to check "create classgroups automatically."

Clicking the "Index" tab in the navigation bar at the top right of the page will show a simple index of the knowledge base.

See more documentation for configuring VIVO, ingesting data, and manually adding data at http://vivoweb.org/support.

Set the Contact Email Address

If you have configured your application to use the "Contact Us" feature (`email.smtpHost` is set in the `runtime.properties` file), you will also need to add an email address to the VIVO application. This is the email to which the contact form will submit. It can be a list server or an individual's email address.

Log in as a system administrator. Navigate to the "Site Admin" table of contents (link in the right side of the header). Go to "Site Information" (under "Site Configuration"). In the "Site Information Editing Form," enter a functional email address in the field "Contact Email Address" and submit the change.

If you set the `email.smtpHost` in the `runtime.properties` file, and do NOT provide an email address in this step, your users will see an error message instead of the expected contact form.

Review the VIVO terms of use

VIVO comes with a "Terms of Use" statement linked from the footer. The "Site Name" you assign in the "Site Information" form under the **Site Admin** area will be inserted into the "Terms of Use" statement. If you want to edit the text content more than just the "Site Name", the file can be found here:

`[vivo_source_dir]/vitro-core/webapp/web/templates/freemarker/body/termsOfUse.ftl`

Your "Terms of Use" statement is also referenced in the Linked Open Data (RDF) that your site produces, so you should be sure that it accurately reflects the way that your data may be used.

Be sure to make the changes in your source files and deploy them to your tomcat so you don't lose your changes next time you deploy for another reason.

Was the installation successful?

If you have completed the previous steps, you have good indications that the installation was successful.

- When you Start tomcat, you see that Tomcat recognizes the webapp, and that the webapp is able to present the initial page.
- When you Log in and add RDF data, you verify that you can log in to the root VIVO account.

The startup status will indicate if the basic configuration of the system was successful. If there were any serious errors, you will see the status screen and will not be allowed to continue with VIVO. If there are warnings, you will see the status screen when you first access VIVO, but after that you may use VIVO without hinderance. In this case, you can review the startup status from **siteAdmin -> Startup status**.

Here is a simple test to see whether the ontology files were loaded:

- Click on the "Index" link on the upper right, below the logo. You should see a "locations" section, with links for "Country" and "Geographic Location." The index is built in a background thread, so on your first login, you may see an empty index instead. Refresh the page periodically to see whether the index will be populated. This may take some time: with VIVO installed on a modest laptop computer, loading the ontology files and building the index took more than 5 minutes from the time that Tomcat was started.
- Click on the "Country" link. You should see an alphabetical list of the countries of the world.

Here is a test to see whether your system is configured to serve linked data:

- Point your browser to the home page of your website, and click the "Log in" link near the upper right corner. Log in with the `rootUser.emailAddress` you set in `runtime.properties`. If this is your first time logging in, you will be prompted to change the password.
- After you have successfully logged in, click "site admin" in the upper right corner. In the drop down under "Data Input" select "Faculty Member (core)" and click the "Add individual of this class" button.
- Enter the name "test individual" under the field "Individual Name," scroll to the bottom, and click "Create New Record." You will be taken to the "Individual Control Panel." Make note of the value of the field "URI" - it will be used in the next step.
- Open a new web browser or browser tab to the page http://marbles.sourceforge.net/. In the pink box on that page enter the URI of the individual you created in the previous step and click "open."
- In the resulting page search for the URI of the "test individual." You should find it towards the bottom of the page next to a red dot followed by "redirect (303)." This indicates that you are successfully serving linked RDF data. If the URI of the "test individual" is followed by "failed (400)" you are not successfully serving linked data.

Finally, test the search index.

- Type the word "Australia" into the search box, and click on the Search button.You should see a page of results, with links to countries that border Australia, individuals that include Australia, and to Australia itself. To trigger the search index, you can log in as a site administrator and go to **Site Admin -> Rebuild search index**.