

System Requirements

- [Hardware Recommendations](#)
 - [Minimum Specification](#)
 - [Recommended Specification](#)
- [Software Requirements](#)
 - [Operating System](#)
 - [glibc 2.14+ \(Linux only\)](#)
 - [Java 8 or 11](#)
 - [Maven 3.6.1 or later](#)
 - [Configuring a Proxy](#)
 - [Tomcat 8 or 9](#)
 - [Optional - MySQL / MariaDB 5.5 or later \(or any other supported by Jena SDB\)](#)

Hardware Recommendations

You can install and run VIVO on most modern PC, laptop, or server hardware. Whilst the application layer needs a reasonable amount of memory, the majority of the workload is placed on the storage layers, which as a semantic web application means the triple store. As VIVO aims to be agnostic to the triple store, the precise requirements will depend on your choice of triple store. However, starting from the VIVO 1.12, the default configuration is to use Jena TDB, while VIVO can be [optionally configured](#) to use MySQL as a backing store for Jena SDB. TDB is another triple store from the Jena project, but TDB keeps its data in directory of flat files, while SDB uses a relational database. It is recommended that you have very high IO bandwidth for the file system used by Jena TDB or MySQL, and significant memory for caching layers of the database engine.

Minimum Specification

2 core x64 processor, 2GB RAM, 100GB HDD

Recommended Specification

4 core x64 processor, 16GB RAM, 500GB SSD

Note: If you plan on using the optional Jena SDB database option, I/O performance for MySQL is critical to the responsiveness of the application. Regardless of chosen database, the fastest SSD you can specify will help, as will having direct (e.g. not virtualised) access to it.

Software Requirements

Operating System

VIVO is largely agnostic to the OS that it is running on - as a Java application, it is dependent on having a Java Virtual Machine and a Tomcat servlet container. It should be possible to install and run VIVO on any OS where you are able to provide all the other software requirements.

However, most sites will run their installations on a Linux server, and you may find that it is easier to follow the installation instructions on a Linux / UNIX variant. Notably, if you are running Windows, you may need to stop running processes (e.g., Tomcat) to complete some of the instructions, due to file locking semantics on Windows.

glibc 2.14+ (Linux only)

Linux users, note that one of VIVO's dependencies requires your OS be running [glibc 2.14+](#). All modern Linux variants satisfy this requirement, however some older releases do not. The table below summarizes the minimum release that includes a compatible* version of glibc. The ability to update glibc to 2.14+ on older OS versions varies by distribution.

Distribution	Ubuntu	Debian	Fedora	CentOS	RHEL	SUSE	Gentoo
Minimum version	12	8	16	7	7	12	12

*This table was created using data from distrowatch.com, accuracy and compatibility not guaranteed. Please leave a comment if you find this table is inaccurate.


Java 8 or 11

The minimum requirement is Java 8. Both OpenJDK and Oracle JVMs are compatible. Other JVMs that meet the JDK 8 specification may work, but have not been tested.

OpenJDK 11 is also supported.

Note that you need to have the full Java Development Kit installed in order for Tomcat to operate correctly - the runtime alone is not sufficient.

Warning

 Java 9 and 10 have not been tested and are not supported

Maven 3.6.1 or later

The installation mechanism uses Maven to package and deploy the VIVO application and other necessary files. Additionally, the development environment also uses Maven to compile the and package the code.

The minimum version of Maven required is 3.6.1, although it is better to use a more recent version of the 3.x releases where possible.

Maven can be downloaded from the following location: <http://maven.apache.org/download.html>, although you may use a version supplied by your operating system / package manager, providing it meets the minimum requirements.

Configuring a Proxy

You can configure a proxy to use for some or all of your HTTP requests in Maven. The username and password are only required if your proxy requires basic authentication (note that later releases may support storing your passwords in a secured keystore, in the mean time, please ensure your *settings.xml* file (usually `$(user.home)/.m2/settings.xml`) is secured with permissions appropriate for your operating system).

Example:

```
<settings>
.
.
<proxies>
  <proxy>
    <active>true</active>
    <protocol>http</protocol>
    <host>proxy.somewhere.com</host>
    <port>8080</port>
    <username>proxyuser</username>
    <password>somepassword</password>
    <nonProxyHosts>www.google.com|*.somewhere.com</nonProxyHosts>
  </proxy>
</proxies>
.
.
</settings>
```

Tomcat 8 or 9

VIVO is a web application, which requires a servlet engine to host. It has been tested with Tomcat 8.5, and 9. The applications make use of Tomcat context.xml configuration files - if you wish to use an alternative servlet engine, you will need to make the appropriate adjustments.

You may use Tomcat as supplied by your operating system / package manager providing it meets the minimum requirements, or you can download it from: <http://tomcat.apache.org/download-85.cgi>. Note, VIVO does not currently support Tomcat 10+.

Tomcat User: When running, Tomcat is usually launched under an unprivileged user account. As VIVO needs to be able to read and write to the home directory, you must ensure that permissions are set on the home directory correctly. This is most easily achieved by assigning ownership to the user that Tomcat is running as.

Optional - MySQL / MariaDB 5.5 or later (or any other supported by Jena SDB)

VIVO can optionally be configured to use a legacy database called Jena SDB, which requires an SQL database to operate. This may be an attractive option for system administrators familiar with SQL databases, however note that Jena SDB was 'retired' by the Apache Jena project in 2020.

Once MySQL is installed, you only need to create a user and schema - see [Jena SDB and MySQL setup](#). VIVO will create the necessary tables and load the default data on startup.

Alternative databases: Jena SDB supports other databases - including PostgreSQL and Oracle. If you wish to use a different database, you will need to add the appropriate Java libraries to the application, and configure the `VitroConnection.DataSource.*` settings in `runtime.properties` so that Jena knows what database it is operating with.