

VIVO Build and Installation Process

As of VIVO 1.12.0, the build and installation process for the application has changed, as detailed on this page.

Recent versions of VIVO required system administrators or anyone installing VIVO to run a Maven build process, which also copied the exploded "web application archive" (war) into the Tomcat "webapps" directory.

The updated process in 1.12.0 separates the process of building the source code into a "war" file from the process of installation into Tomcat (or Jetty, now!).

One implication of this change is that each release of VIVO will include a deployable "war" file... no building of the software is required to test and/or use VIVO.

Additionally, for those who are used to customizing VIVO with a "custom installer", the same functionality is still available by creating a GitHub fork of [vivo-custom-webapp](#). You can then place your customized code, RDF, property, and template files into your own vivo-custom-webapp. When you build your vivo-custom-webapp, the core VIVO war file is used as a base into which all customizations are included with the Maven overlay process. The resultant war file can then be installed into your servlet container.

VIVO Home Directory

Packaged within the VIVO war file are the RDF and config files required by the application. When VIVO starts-up, those files are written to the "VIVO home" directory.

Subsequent restarts of the application will not overwrite these files unless:

- the files in VIVO home have not changed since the last time the VIVO application was started, and
- there is an updated version of one of the VIVO home files in the VIVO war file.

There are three mechanisms for specifying where the "VIVO home" location for the application to use:

1. If building VIVO from source, you may specify the location by including the "vivo-dir" parameter, for example:

```
a. mvn clean install -Dvivo-dir=/opt/vivo/home
```

2. Or, you may specify the location by defining the "vivo-dir" option in the servlet container's (e.g. Tomcat) options, for example:

a. **Potentially in "/etc/defaults/tomcat9"**

```
JAVA_OPTS="$ {JAVA_OPTS} -Dvivo-dir=/opt/vivo/home"
```

3. Thirdly, particularly in the case where you have multiple instances of VIVO deployed in the same servlet container, you can specify the location with a context file within your servlet container's configuration. An [example context.xml](#) file for Tomcat is included in the source. Where to place the context file for Tomcat is detailed in the [Tomcat documentation](#).

VIVO will place the following three files in the "VIVO home" directory on start-up:

- default.runtime.properties
- default.applicationSetup.n3
- default.developer.properties

As the names imply, these files contain default configuration. If you want to change your configurations (which you likely will), copy any or all of those three files and remove the "default." prefix. Files named, "runtime.properties", "applicationSetup.n3", and "developer.properties" will take precedence over the default configurations. **Note:** If you make non-default configurations, make sure that the files are readable by the system user under which the Tomcat or Jetty service is being run.

Digest File

In order to manage the state of VIVO home directory a digest.md5 file has been introduced. It will afford maintaining customizations through restarts and upgrades. The implementation can be found in [VivoHomeDirectory.java](#).

On first startup

- packed VIVO home tar file is untarred into vivo home
 - if file already exists it will be overwritten*
- digest.md5 file is created of a checksum per tar entry

On restart

- digest.md5 is read in to determine if any files have changed
- packed VIVO home tar file is untarred
 - if file already exists and matches digest checksum and does not match tar entry checksum, file is overwritten
- digest.md5 file is created of a checksum per tar entry

*if it is desired to keep VIVO home directory customizations on upgrade from previous version the following command can generate an initial digest.md5

```
find /vivo/home -type f | cut -c3- | grep -E '^bin/|^config/|^rdf/' | xargs md5sum > /vivo/home/digest.md5
```

Application Name

By default, the VIVO software uses "vivo" as its application name. This name is used in two places:

1. The name of the war file that is produced in the build process, and
2. The name of the application log file, e.g. "vivo.all.log"

The three mechanisms mentioned above in the "VIVO Home Directory" section can be used to set the application name. The parameter name to be used is:

- *app-name*

For example:

```
mvn clean install -Dvivo-dir=/opt/vivo/home -Dapp-name=vivo2
```

Installation Process

Pre-conditions

Before installing VIVO, you must ensure that a directory exists that will serve as the VIVO application's configuration and storage location. In addition to existing, this directory must be readable and writable by the system user under which the servlet container service is being run.

In addition to a readable/writable "VIVO home" directory, you must also [install and configure Solr](#).

Once those two pieces are in place, installation simply involves deploying the VIVO war file into the servlet container (Tomcat or Jetty). This can be done through the servlet container's web interface, HTTP API, or by copying the VIVO war file into the location specified by the servlet container. For example with Tomcat: `$CATALINA_HOME/webapps/`

Build Process

If building VIVO from source code, you can use the standard Maven build command:

```
mvn clean install
```

This will produce a VIVO war file in the "**installer/webapp/target/**" directory, named with the default application name: *vivo.war*

The build process also creates an intermediate war file in the "webapp/target" directory. This war file does not contain the complete VIVO application and will not run if copied into Tomcat's webapps directory. Be sure that you use only the war in "installer/webapp/target".

If you do not specify the "*vivo-dir*" location as a build parameter, you must provide the "*vivo-dir*" location with one of the other two detailed in the "VIVO Home Directory" section above.

The only two VIVO-specific build parameters are:

- `-Dvivo-dir=some-directory-location` (no default)
- `-Dapp-name=some-name` (defaults to: "vivo")

Customizations

If you want to customize your VIVO application using Maven's overlay mechanism at build-time, please see: [vivo-custom-webapp](#)