

Developers' installation: obtaining VIVO source code from Git

- [VIVO code and Vitro code](#)
 - [Working from the release files](#)
 - [Working from the GitHub repositories](#)
- [Revision Information](#)
 - [Working from the release files](#)
 - [Working from the GitHub repositories](#)
- [A Frequently Asked Question](#)

Some people prefer to work with the very latest source code.

Some people prefer to get the VIVO code directly from the GitHub repositories, instead of downloading the distributed release files. The `master` branch of the repositories always contains the latest release code, and is available for anyone to access.

If you prefer not to wait for the next stable release, you can get the VIVO code as it is being developed. The `develop` branch has not been fully tested and is not recommended for production use, but it can be useful for people who want a "close-up" look at how VIVO is progressing.

VIVO code and Vitro code

The principal difference between the release files and the repositories is the location of the Vitro code, in relation to the VIVO code.

Working from the release files

In the VIVO release files, the Vitro code is embedded within VIVO.

When you create the `build.properties` file, you set `vitro.core.dir` accordingly.

Working from the GitHub repositories

When you use the code from the repositories, it is customary to place the Vitro code beside the VIVO code.

When you create the `build.properties` file, `vitro.core.dir` must be set to reflect this difference.

Working from the release files	Working from the GitHub repositories
<pre>/Users/jeb228/vivo-rel-1.7 build.xml config doc example.build.properties languages lib productMods rdf README.md revisionInfo src test themes utilities vitro-core doc opensocial README.md revisionInfo solr utilities webapp</pre>	<pre>/Users/jeb228/git Vitro doc opensocial README.md solr utilities webapp VIVO build.xml config doc example.build.properties languages lib productMods rdf README.md src test themes utilities</pre>
build.properties	build.properties

<code>vitro.core.dir = ../vitro-core</code>	<code>vitro.core.dir = ../Vitro</code>
---	--

Revision Information

Working from the release files

In the VIVO release files, the `VIVO` directory contains a file called `revisionInfo`. This file states the release version of VIVO, and a seven-character hashcode that can be used to locate the release tag in the GitHub repository. The `VIVO/vitro-core` directory contains a similar file, with similar information about the Vitro release.

Working from the GitHub repositories

The `revisionInfo` files are not stored in the GitHub repositories. Instead, when you run the build script, queries are made to Git to find the most recent tag, and the hashcode of the most recent commit. In this way, VIVO can report the level of code that was used to build it.

A Frequently Asked Question

Q: I'm running VIVO in Tomcat on a MacIntosh. When I upload an image file, a strange icon appears in the dock. What's up with that?

A: The image processing code in VIVO uses `javax.imageio.ImageIO`, which in turn uses code in the `java.awt` package. By default, this package will open a work area on your screen, producing this icon even though no visible window is created.

You can prevent this by editing `{tomcat}/conf/setenv.sh`, and adding the `"java.awt.headless"` option. For example,

```
export CATALINA_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=64m -Djava.awt.headless=true"
```