

Building VIVO in 3 tiers

- [Development](#)
- [Deployment](#)
- [Project template](#)

Add a third layer to the VIVO distribution, to keep all of your modifications in one place.

The three tiered build is very simple. In a standard two tier build VIVO replaces anything that is in Vitro as it adds its own special files to the mix. In a three tier we replace anything in Vitro or VIVO with the files that we want to add or modify for our version of VIVO. This allows us to point at newer VIVO versions without modifying the files within.

The files that differ from a regular VIVO build (other than the ones you are adding and changing) are the build.xml and deploy.properties file. I've attached the files [here](#)

- [build.xml](#)
- [deploy.properties](#)

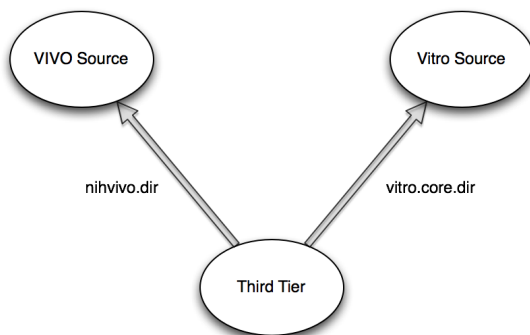
The main difference in these files is the presence of nihvivo.dir. This attribute points to where VIVO is for the build. There are two standard ways to setup VIVO, for development or deployment.

Development

With development you want to look at your VIVO against the VIVO and Vitro sources. this means getting from the git repository the source for each project

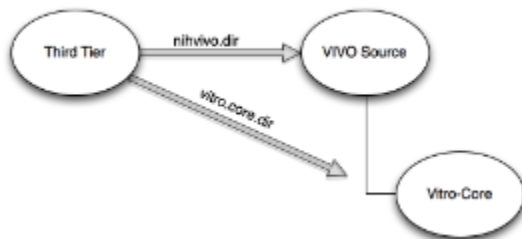
- <https://github.com/vivo-project/Vitro>
- <https://github.com/vivo-project/VIVO>

Your deploy.properties then points at the individual repositories. When you build it pulls the three repositories together and sends them as a single package to Tomcat.



Deployment

In deployment you have just your third tier and the released source for VIVO. In that released source is the vitro code that goes with the latest release of VIVO. Your deploy.properties for a deployment build will need to point to the internal vitro-core folder for the vitro code.



Project template

If you would like to get started with the three tiered build process, there is a [project template on Github](#) that includes the necessary build.xml, deploy.properties and directory structure. This template uses [Git submodules](#) to pull in VIVO and Vitro from the main vivo-project repository.