

# Changing the appearance of VIVO

- [Introduction](#)
  - [Making changes to VIVO](#)
    - What is included here?
    - What is not included here?
  - [VIVO is already customized](#)
    - VIVO and Vitro
    - How VIVO is merged into Vitro
  - [Adding your own customizations](#)
    - Working in the GUI
    - RDF files
    - Changes to the source files
    - A third tier?
  - [Tool summary](#)
    - Required skills
    - The tools

## Introduction

### Making changes to VIVO

The VIVO application is a popular tool for research networking. Most VIVO sites put their own changes into VIVO, in order to create a distinctive appearance, or to satisfy their particular needs.

VIVO supports an assortment of tools and techniques for making these changes. Some changes can be accomplished while VIVO is running, simply by setting values on a form. Other changes require you to add or modify configuration files that control the application. Still other changes are accomplished by editing the VIVO code, re-building, and re-deploying the application.

#### What is included here?

This document describes the most common ways of modifying VIVO. The changes affect the appearance, layout, and content of the pages in the application.

#### What is not included here?

- Data operations
  - VIVO is only as good as the data it holds, and how the data is structured. The task of populating VIVO with data is very different on each site. Techniques for ingesting data into VIVO are covered in a separate document.
- The ontology
  - VIVO recognizes classes, instances, and properties base on the statements in its ontology. To a large extent, this determines how VIVO behaves. This document does not discuss changes to the ontology.
- The Java code
  - VIVO is open source software, so all changes are permitted. However, changes to the Java code are not discussed here. The sole exception is the custom editing forms, which do require some Java.
- The search index
  - The search index in VIVO can be configured to exclude certain classes of individuals. It can also be customized to include additional data fields. This document does not discuss how to customize the search index.
- The supporting technologies
  - VIVO relies on a triple-store and a search engine. It can be used with an external authentication system. Options for configuring or changing these technologies are discussed in the installation guide.

### VIVO is already customized

Customization is built in to the heart of VIVO. VIVO itself is a customization of a more basic product called Vitro.

Here is how Vitro has been customized to become VIVO

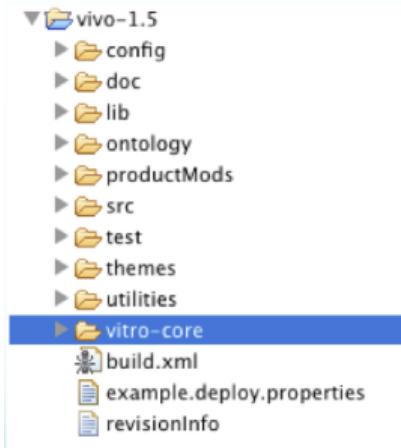
Vitro	VIVO
No ontology	Includes an ontology for Research Networking
Minimal theme	Rich theme.
Default display rules	Annotations are used to: <ul style="list-style-type: none"><li>• Assign data properties to groups</li><li>• Arrange property groups on the page</li></ul>

Default permissions	Display and editing permissions are customized, based on the ontology
Default editing forms	Editing is customized to the ontology
Default search index	Search index contains additional fields, specific to VIVO
Default functionality	Additional functionality: visualizations, interface to Harvester, QR codes, etc.
<b>In total:</b> A general-purpose tool for working with Semantic Data.	<b>In total:</b> A specialized tool for Research Networking

## VIVO and Vitro

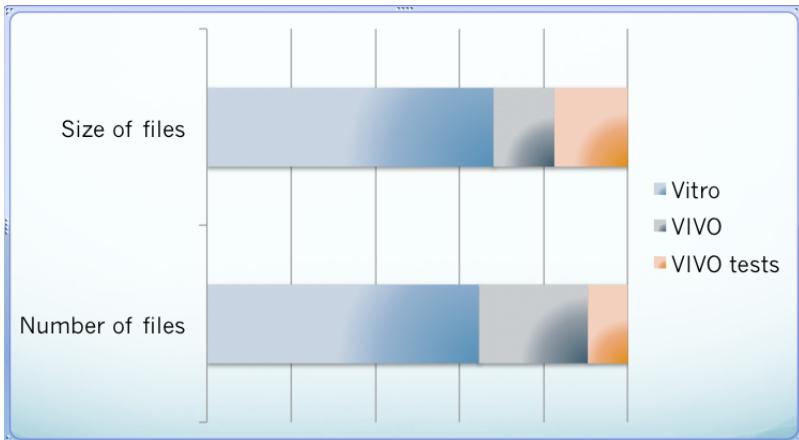
### VIVO includes Vitro

When the VIVO distribution files are unpacked, the Vitro files are inside the main directory



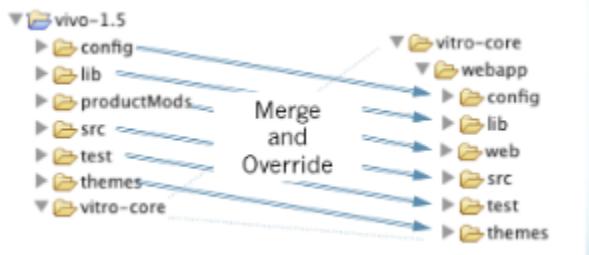
### Most of VIVO is Vitro

This graphic from VIVO release 1.5 shows how much of VIVO is actually Vitro. Vitro makes up more than 60 percent of the VIVO distribution. If you were to remove the VIVO acceptance tests, about 80 percent of VIVO would actually be Vitro.

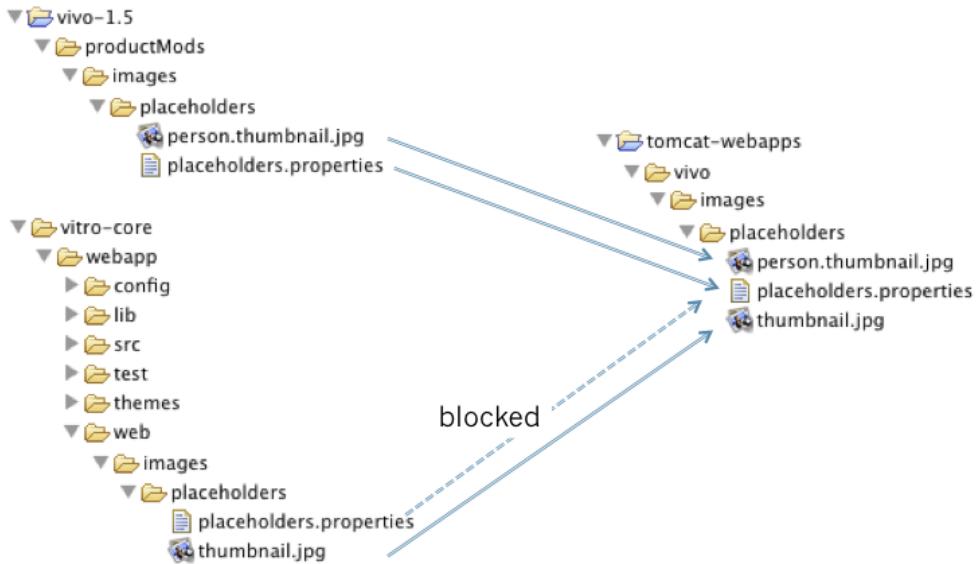


### How VIVO is merged into Vitro

The build process in VIVO begins by overlaying the VIVO directory structure on the Vitro directory structure.



If a file in VIVO has the same name and directory path as a file in Vitro, the file in VIVO will replace (override) the file in Vitro, which is blocked. VIVO files that do not override Vitro files are added to the merged directories. The build process continues by compiling, testing, integrating and packaging the merged directories.



## Adding your own customizations

How do you add your changes to VIVO? Perhaps more important, how do you keep your changes when you upgrade to a newer release of VIVO?

## Working in the GUI

When you use forms in VIVO, the values you enter are kept in the triple-store. They will be retained when you upgrade to a new release. If the new release uses a different format to store the values, your changes will be migrated to the new format.

## RDF files

Some customizations require that you add or modify an RDF file in your VIVO home directory. In general, it's best to create a new file to contain the RDF statements, so you can easily carry your changes to a new VIVO release.

A "clean" build of VIVO will erase the RDF files in your VIVO home directory. You will need to re-create these files after the migration.

## **Changes to the source files**

As with the RDF files, you should favor new files over changes to existing files. This will make it easier to carry your changes to a new release.

## A third tier?

The discussion of VIVO and Vitro shows how the two code bases are combined during the build process. Some VIVO sites keep their local modifications in a third layer. This layer (or tier) is combined with VIVO and Vitro during the build.



The three-tier approach is a little harder to set up than the standard build, but it has the advantage of keeping all of your local modifications separate from the VIVO distribution. When the time comes to upgrade to a new release, there is no question about what files contain your local changes.

The two-tier build is fine, if your changes are limited to:

- Annotations on the ontology
- A custom theme
- Page management
- Language support

Three-tiers might be better if you will be using

- Custom list views
- Custom short views
- Custom entry forms
- Java changes

The VIVO Installation Instructions contain more details about how to set three tiers and there is a [Git project template available](#).

## Tool summary

### Required skills

The customization tools require different levels of knowledge. Some are as simple as filling out a web form. Most require the ability to write HTML, with additions from the Freemarker template engine. Some require Java programming.

As the tools are described, these terms will be used to specify the skills needed:

	<b>Knowledge required</b>
<b>Basic</b>	Requires an understanding of VIVO concepts.
<b>Web development</b>	The usual technologies for writing web sites, including HTML, CSS, and JavaScript. Knowledge of the Freemarker template engine.
<b>RDF</b>	Modify or create RDF data files, using RDF/XML, Turtle, or N3 format.
<b>SPARQL</b>	Create queries against the triple-store, using SPARQL.
<b>Java</b>	Create or modify Java code.
<b>OpenSocial</b>	Create or modify OpenSocial gadgets, written in JavaScript.

## The tools

	<b>What does it do?</b>	<b>How?</b>	<b>Required skills</b>
<a href="#">Creating a custom theme</a>	Create your own "brand" for VIVO. <ul style="list-style-type: none"> <li>• Change colors, logo, headings, footers, and more.</li> </ul>	CSS files, JavaScript files, and templates for HTML.	Web development

<a href="#">Annotations on the ontology</a>	Control how data is displayed. <ul style="list-style-type: none"><li>Property groups, labels, display order, hidden properties, and more.</li></ul>	Interactive.	Basic
<a href="#">Home page customizations</a>	Choose from home page options. <ul style="list-style-type: none"><li>Add a geographic focus map.</li></ul>	Edit your home page template to include a selection of sub-templates.	Web development
<a href="#">Page management</a>	Add new pages to VIVO. <ul style="list-style-type: none"><li>Static pages, navigation pages, or dynamic reports.</li></ul>	Interactive.	Web development, optional SPARQL
<a href="#">Profiles for classes</a>	Use one type of profile page for people and another for organizations.	Create page templates. Configure VIVO to associate them with classes.	Web development, RDF
<a href="#">Multiple profile types for foaf: Person</a>	Provide a choice of formats for profile pages. <ul style="list-style-type: none"><li>Each page owner selects the format for his own page.</li></ul>	Edit page templates. Perhaps connect to a Website image capture service.	Web development
<a href="#">Enriching profile pages with SPARQL queries</a>	Display additional data on a profile page.	Write a SPARQL query. Create a template to display the results. Configure VIVO to use it.	Web development, SPARQL, RDF
<a href="#">Enhancing page templates with SPARQL queries</a>	Display additional data in any page template.	Write a SPARQL query. Modify a template to display the results. Configure VIVO to use it.	Web development, SPARQL, RDF
<a href="#">Custom list views</a>	Change how certain properties are displayed <ul style="list-style-type: none"><li>Change the layout for that property</li><li>Display additional data with each value.</li></ul>	Write a SPARQL query. Create a template to display the results. Configure VIVO to use it.	Web development, SPARQL, RDF
<a href="#">Custom short views</a>	Change how search results are displayed <ul style="list-style-type: none"><li>Display depends on the type of result (Person, Document, etc.).</li></ul> Also change display on index pages and browse pages.	Write a SPARQL query. Create a template to display the results. Configure VIVO to use it.	Web development, SPARQL, RDF
<a href="#">Custom entry forms</a>	Create data entry forms <ul style="list-style-type: none"><li>Add or edit complex data structures.</li></ul>	Write a generator class in Java. Create a template for the editing form.	Web development, SPARQL, RDF, Java
<a href="#">Using Open Social Gadgets</a>	Create optional content for profile pages. <ul style="list-style-type: none"><li>Each page owner configures the gadgets for his own page.</li></ul>	Create gadgets from JavaScript, or install existing gadgets.	Web development, OpenSocial
<a href="#">Language support</a>	Languages other than English <ul style="list-style-type: none"><li>Use VIVO in Spanish</li><li>Allow viewers to choose their preferred language.</li><li>Implement other languages.</li></ul>	Create files of phrases in the desired language, or install existing files.	Basic