

Customizing the Interface

- [Introduction](#)
 - [Making changes to VIVO](#)
 - [VIVO is already customized](#)
- [Adding your own customizations](#)
 - [Working in the GUI](#)
 - [RDF files](#)
 - [Changes to the source files](#)
- [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.

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">• Assign data properties to groups• Arrange property groups on the page
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.
In total: A general-purpose tool for working with Semantic Data.	In total: A specialized tool for Research Networking

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.

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:

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

The tools

	What does it do?	How?	Required skills
Creating a custom theme	Create your own "brand" for VIVO. <ul style="list-style-type: none">Change colors, logo, headings, footers, and more.	CSS files, JavaScript files, and templates for HTML.	Web development
Annotations on the ontology	Control how data is displayed. <ul style="list-style-type: none">Property groups, labels, display order, hidden properties, and more.	Interactive.	Basic
Home page customizations	Choose from home page options. <ul style="list-style-type: none">Add a geographic focus map.	Edit your home page template to include a selection of sub-templates.	Web development
Menu and page management	Add new pages to VIVO. <ul style="list-style-type: none">Static pages, navigation pages, or dynamic reports.	Interactive.	Web development, optional SPARQL
Profiles for classes	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
Multiple profile types for foaf: Person	Provide a choice of formats for profile pages. <ul style="list-style-type: none">Each page owner selects the format for his own page.	Edit page templates. Perhaps connect to a Website image capture service.	Web development
Enriching profile pages with SPARQL queries	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

Enhancing page templates with SPARQL queries	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
Custom list views	Change how certain properties are displayed <ul style="list-style-type: none"> • Change the layout for that property • Display additional data with each value. 	Write a SPARQL query. Create a template to display the results. Configure VIVO to use it.	Web development, SPARQL, RDF
Custom short views	Change how search results are displayed <ul style="list-style-type: none"> • Display depends on the type of result (Person, Document, etc.). 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
Custom entry forms	Create data entry forms <ul style="list-style-type: none"> • Add or edit complex data structures. 	Write a generator class in Java. Create a template for the editing form.	Web development, SPARQL, RDF, Java
Using Open Social Gadgets	Create optional content for profile pages. <ul style="list-style-type: none"> • Each page owner configures the gadgets for his own page. 	Create gadgets from JavaScript, or install existing gadgets.	Web development, OpenSocial
Language support	Languages other than English <ul style="list-style-type: none"> • Use VIVO in Spanish • Allow viewers to choose their preferred language. • Implement other languages. 	Create files of phrases in the desired language, or install existing files.	Basic