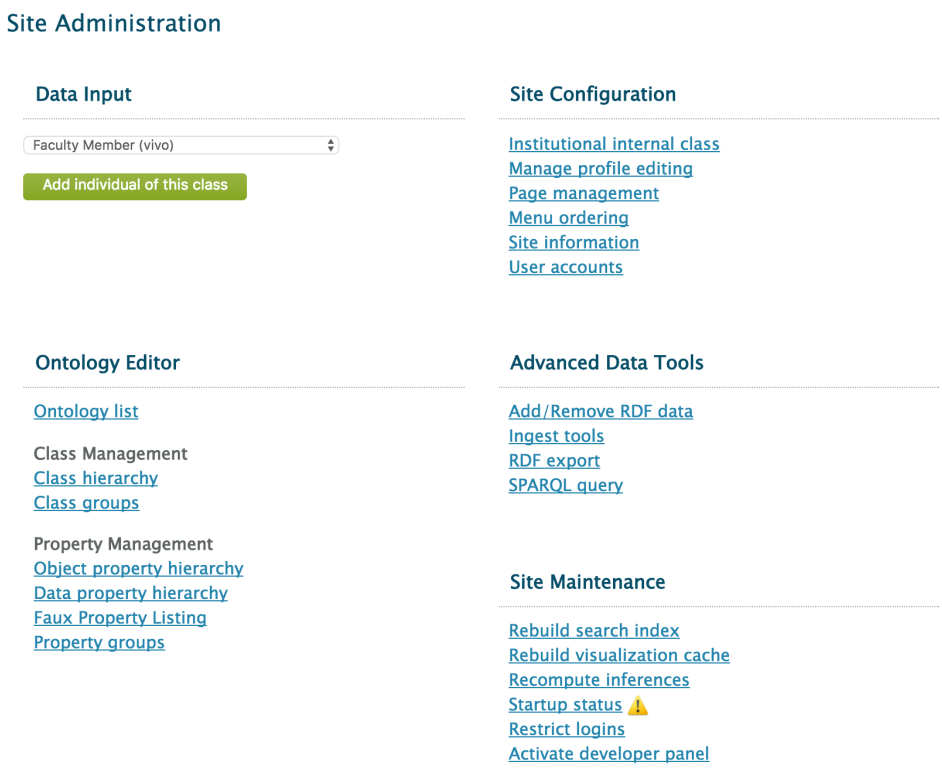
The Site Administration Page

Site Administration

Once you are logged into VIVO, you will notice in the upper right hand portion of the page links to "Index" and "Site Admin", alongside a drop-down menu with your name on it, and containing links to "My account" and "Log out".

Once you have logged into VIVO, clicking on the “Site Admin” link takes you to the “Site Administration” page. As an administrator, you will be able to access all five feature and content areas of VIVO: Data Input, Ontology Editor, Site Configuration, Advanced Data Tools, and Site Maintenance. Each is introduced below.



Data Input

There are three ways to manually input data into VIVO. 1) on the Site Administration Menu, a new individual of any class may be added directly through the Data Input menu. 2) Selections can be made on many of the pages to add individuals. For example, on user profile pages, users with editing privileges can add, change and remove data. 3) Data can be added as a batch, a collection of RDF triples in a format known to VIVO and expressed in the ontologies known to VIVO.

On the Site Administration page, you can enter a new individual of any type by selecting the type from the drop down menu, and pressing "Add Individual of this class." VIVO will add an individual of the class you have selected, creating a new URI for the individual, and creating an assertion that the individual is a member of the class you have selected. Once an individual has been created, object and data properties may be added for that individual on the page displaying the individual’s profile. The object and data properties presented for editing will vary by the type of the individual, in accordance with the ontology;

Ontology Editor

In VIVO, information is identified by references to Unique Resource Identifiers (URIs). URIs can be used by other web pages and applications to locate and retrieve specific chunks of data. The detailed level to which VIVO captures information enables complex relationships among data to be represented.

The VIVO web application is built using RDF "triples" or statements consisting of a subject (known as an individual, item, or entity), a predicate (an object property or a data property) and an object (any individual in VIVO). Subject-predicate-object triples express the relationships among the individuals in VIVO via object properties and support attributes of individuals via data properties.

The first two parts (subject and predicate) of every triple are URIs. An object property triple has the URI of another individual in VIVO its object, while the third element of a data property triple is a data value – typically a text string, number, or date.

Ontology List - VIVO supports keeping an internal list of ontology namespaces and corresponding prefixes to facilitate using external ontologies as well as to help differentiate local ontology additions from VIVO core.

Class Management

Individuals in VIVO are typed as members of one or more classes organized and displayed as a hierarchy.

Class hierarchy - The class hierarchy provides a framework to help identify the different types of individuals modeled in a VIVO application. In the Class Hierarchy page, you can edit/add classes, add entities to a class, and add auto links.

Class groups - Class groups are a VIVO-specific extension to support using VIVO as a public website as well as an ontology and content editor. Class groups are a means to organize the classes in VIVO into groups. They represent the facets seen when VIVO is searched (people, activities, events, organizations, etc).

Property Management

If classes define what each individual in VIVO is, properties define how that individual relates to other individuals and allow an individual to have attributes of its own. VIVO has two property editors, one for object properties and another for data properties.

Object property hierarchy - Object properties represent the relationship between entities (also known as items or individuals) in VIVO. Object properties can be created and edited from the Object Property Hierarchy.

Data property hierarchy - A data property connects a single subject individual (e.g., a Person or Event) with some form of attribute data. Data properties can be created and edited from the Data property hierarchy link.

Faux property listing - Faux properties are a VIVO-specific extension to allow the same object property to be used in various contexts, with a context specific label and context specific domain and range. A listing of the faux properties in VIVO. You can display the list alphabetically, or organized by base property. The Site Administration page provides a simple view-only listing. See Create and edit faux properties to manage faux properties.

Property groups - Like class groups, property groups are a VIVO-specific extension to support using VIVO as a public website as well as an ontology and content editor.

Site Configuration

This section discusses the site configuration aspects of VIVO. It enables administrators to add or adjust to their institution's site specific details, as well as to manage menus, tabs, and user accounts.

Institutional internal class – set the class that will be used to indicate that individuals are part of your institution. See Create, Assign, and Use an Institutional Internal Class

Manage profile editing – Assign profile editors to individual profiles. Use this feature to allow someone other than the profile owner to edit the owner's profile.

Page management – Create and manage custom pages, as well the presence of pages on menus. See Menu and page management

Menu Ordering – order the menus on the main VIVO navigation banner. See Menu and page management

Site Information

The Site information link provides administrators with the capabilities of editing and adding site specific details for that institution's instance of VIVO.

Site Name — Text entered here will be displayed in the browser title bar and bookmark label. It is set to "VIVO" by default.

Contact email address — This field is the email address or listserv that you want the Contact Us form to use. The SMTP host in your configuration file (runtime.properties) must be set for the Contact Us form to work as intended.

Theme — The default theme is "wilma". If you create a new theme (see Creating a custom theme), then it should be available to choose in this drop-down pick list.

Copyright text — Text entered here for a label in the footer for the copyright URL.

Copyright URL — The URL you want the copyright to go to in the footer. It could be your institution's copyright information or the actual institution.

Advanced Tools

The Advanced tools are VIVO's built-in features for data management and export. Please refer to the Advanced Tools section below for detailed instructions.

In addition, many VIVO adopters may require additional information regarding the importing and exporting of RDF data and creating SPARQL queries.

There are several avenues available to acquire guidance with these advanced tools. Information sources such as the VIVO Data Ingest Guide, the W3C's Resource Description Framework model, and the W3C's SPARQL Query Language for RDF, to name a few. Please refer to Appendix A for links and additional resources.

Add/Remove RDF data **–** This tool allows for the manipulation of RDF data in the main model through importing RDF documents for addition or removal.

Ingest tools – A suite of data management tools. See below for a description of each tool.

RDF export - This tool allows for the export of ontology and data in a variety of RDF formats. Options include:

- Export all instance data
- Export a specific ontology such as FOAF, VIVO core, SKOS, etc.
- Export the entire ontology for VIVO

SPARQL query - This tool allows SPARQL select, construct, and describe statements against the main model to be saved in a variety of formats including: CSV, RDF/XML, N3 and more.

Ingest tools

Manage Jena Models – This tool allows for the management of the main webapp, as well as separate data models and datasets. The ability to attach separate models to the webapp, load RDF data to a mode, clear statements, and output models as N3 RDF is performed here.

Subtract One Model from Another — This tool allows for the comparison of models for updating information that already exists in VIVO. By subtraction of a current model from a newly constructed model (from the same data source) and vice versa, the additions and subtractions for updating the data are generated.

Convert CSV to RDF — This tool allows for VIVO to read and convert CSV (comma-separated values) and Tab-delimited data into RDF

Convert XML to RDF — This tool allows for VIVO to read and convert well-formed XML into RDF

Execute SPARQL CONSTRUCT — This tool allows for using SPARQL to produce desired RDF from one or multiple source models. This tool is commonly used to map classes and properties to VIVO namespace(s).

Generate Tbox — Tbox statements describe the terms of controlled vocabularies, for example, a set of classes and properties that constitute the ontology. This tool allows for the creation of a Tbox from one or multiple source models.

Name Blank Nodes — This action turns blank nodes, a node in an RDF graph which is not identified by a URI and is not a literal, into nodes with either randomly generated or pattern based URIs.

Smush Resources — This tool allows for using a compression method to distinguish like entities and "Smush" them together based on the specified URI of a property.

Merge Resources — This tool allows two individuals with different URIs to be collapsed into a single URI. Any statements using the "duplicate individual URI" will be rewritten using the "primary individual URI." If there are multiple statements for a property that can have only a single value, the extra statements will be retracted from the model and offered for download.

Process Property Value Strings — This tool allows for an arbitrary method on a Java class available on the application class path to transform string values of a given property. The method should take a single String as a parameter and return a String.

Change Namespace of Resources — This tool will change all resources in the supplied "old namespace" to be in the "new namespace." Additionally, the local names will be updated to follow the established "n" + random integer naming convention.

Split Property Value Strings into Multiple Property Values — This tool allows for parsing multiple property values from a single ingested string. This can be used to parse MeSH Terms, controlled vocabulary, and keywords associated with the ingested data.

Execute Workflow — This tool allows for a simple way of scripting actions (specified in RDF) that would otherwise require manual interaction with the ingest tools.

Dump or restore the knowledge base – dump or restore configuration models or content models

Site Maintenance

Rebuild search index – in some situations, you may need to rebuild the SOLR search index. See Inferences and Indexing

Rebuild visualization cache – Large-scale visualizations like the Temporal Graph or the Map of Science involve calculating total counts of publications or of grants for some entity. Since this means checking also through all of its sub-entities, the underlying queries can be both memory-intensive and time-consuming. For a faster user experience, we wish to save the results of these queries for later re-use. To this end we have devised a caching solution which will retain information about the hierarchy of organizations-namely, which publications are attributed to which organizations-by storing the RDF model. We're currently caching these models in memory. The cache is built (only once) on the first user request after a server restart. Because of this, the same model will be served until the next restart. This means that the data in these models may become stale depending upon when it was last created. To avoid restarting the server in order to refresh the cache, administrators can use the Rebuild visualization cache link.

Recompute inferences – in some cases, you may wish to recompute the inferences in VIVO. See Inferences and Indexing

Startup status – shows the messages that were produced during VIVO startup.

Restrict logins – toggles user login. When logins are restricted, only the root user may login

Activate Developer panel – Shows the developer panel from which additional debugging information is available. See Tips for Interface Developers