

Configuration Reference

- [Overview](#)
- [VIVO Runtime Properties](#)

Overview

VIVO's operation can be determined by setting corresponding properties in runtime.properties.

VIVO Runtime Properties

Property	Description
<code>Vitro.defaultNamespace = http://vivo.mydomain.edu/individual/</code>	This namespace will be used when generating URIs for objects created in the editor. In order to serve linked data, the default namespace must be composed as follows (optional elements in parentheses): scheme + server_name (+ port) (+ servlet_context) + "/individual/" For example, Cornell's default namespace is: http://vivo.cornell.edu/individual/
<code>rootUser.emailAddress = vivo_root@mydomain.edu</code>	The email address of the root user for the VIVO application. The password for this user is initially set to "rootPassword", but you will be asked to change the password the first time you log in.
<code>VitroConnection.DataSource.url = jdbc:mysql://localhost/vitrodb</code> <code>VitroConnection.DataSource.username = vitrodbUsername</code> <code>VitroConnection.DataSource.password = vitrodbPassword</code>	The basic parameters for a database connection. Change the end of the URL to reflect your database name (if it is not "vitrodb"). Change the username and password to match the authorized database user you created.
<code>email.smtpHost = smtp.mydomain.edu</code> <code>email.replyTo = vivoAdmin@mydomain.edu</code>	Email parameters which VIVO can use to send mail. If these are left empty, the "Contact Us" form will be disabled and users will not be notified of changes to their accounts.
<code>vitro.local.solr.url = http://localhost:8080/vivosolr</code>	URL of Solr context used in local VIVO search. This will usually consist of: scheme + server_name + port + vivo_webapp_name + "solr" In the standard installation, the Solr context will be on the same server as VIVO, and in the same Tomcat instance. The path will be the VIVO webapp.name (specified in build.properties) + "solr" Example: vitro.local.solr.url = http://localhost:8080/vivosolr

<pre>selfEditing. idMatching Property = http:// /vivo. mydomain. edu /ns#networkId</pre>	<p>How is a logged-in user associated with a particular Individual? One way is for the Individual to have a property whose value is the username of the user. This value should be the URI for that property.</p>
<pre>externalAuth. netIdHeaderName = remote_userID</pre>	<p>If an external authentication system such as Shibboleth or CUWebAuth is to be used, this property says which HTTP header will contain the user ID from the authentication system. If such a system is not to be used, leave this commented out. See Using an external authentication system</p>
<pre>VitroConnection. DataSource .pool. maxActive = 40</pre>	<p>The maximum number of active connections in the database connection pool. Increase this value to support a greater number of concurrent page requests.</p>
<pre>VitroConnection. DataSource .pool. maxIdle = 10</pre>	<p>The maximum number of database connections that will be allowed to remain idle in the connection pool. Default is 25% of the maximum number of active connections.</p>
<pre>VitroConnection. DataSource .dbtype = MySQL VitroConnection. DataSource .driver = com.mysql. jdbc. Driver VitroConnection. DataSource .validation Query = SELECT 1</pre>	<p>Parameters to change in order to use VIVO with a database other than MySQL. These parameters allow you to change the relational database that is used as the back end for Jena SDB. If you want to use a triple store other than SDB, you will need to edit applicationSetup.n3. See the installation instructions for more details.</p>
<pre>OpenSocial .shindigURL = http://localhost: 8080 /shindigoring</pre>	<p>For OpenSocial integration, the base URL of the ORNG Shindig server. Usually, this is the same host and port number as VIVO itself, with a context path of "shindigoring".</p>
<pre>OpenSocial .tokenService = myhost. mydomain. edu:8777</pre>	<p>For OpenSocial integration, The host name and port number of the service that provides security tokens for VIVO and Shindig to share. For now, the host name must be the actual host, not "localhost" or "127.0.0.1" The port number must be 8777</p>

<pre> OpenSocial .tokenKeyFile = /usr /local /vivo/data /shindig /openssl /securitytokenkey.txt </pre>	<p>For OpenSocial integration. The path to the key file that will be used when generating security tokens for VIVO and shindig to share.</p>
<pre> OpenSocial .sandbox = True </pre>	<p>For OpenSocial integration. Only set sandbox to True for dev/test environments. Comment out or set to False in production</p>
<pre> RDFSService .languageFilter = false </pre>	<p>Show only the most appropriate data values based on the Accept-Language header supplied by the browser. Default is false if not set.</p>
<pre> languages .forceLocale = en_US </pre>	<p>Force VIVO to use a specific language or Locale instead of those specified by the browser. This affects RDF data retrieved from the model, if RDFSService.languageFilter is true. This also affects the text of pages that have been modified to support multiple languages.</p>
<pre> languages .selectableLocales = en_US, es_GO </pre>	<p>A list of supported languages or Locales that the user may choose to use instead of the one specified by the browser. Selection images must be available in the i18n/images directory of the theme. This affects RDF data retrieved from the model, if RDFSService.languageFilter is true. This also affects the text of pages that have been modified to support multiple languages. This should not be used with languages.forceLocale, which will override it.</p>
<pre> orcid .clientId = 0000-0000-0000-000X orcid .clientPassword = 00000000-0000-0000-0000-0000000000 orcid .webappBaseUrl = http://localhost:8080/vivo orcid .messageVersion = 1.0.23 orcid .externalIdCommonName = VIVO Cornell Identifier </pre>	<p>ORCID integration parameters. See Activating the ORCID integration</p>

<pre> orcid. publicApiB aseUrl = h ttp://pub. sandbox. orcid.org /v1.1 orcid. authorized ApiBaseUrl = http://a pi. sandbox. orcid.org /v1.1 orcid. oauthAutho rizeUrl = http://san dbox. orcid.org /oauth /authorize orcid. oauthToken Url = http ://api. sandbox. orcid.org /oauth /token </pre>	<p>Setup for the ORCID sandbox</p>
<pre> orcid. publicApiB aseUrl = h ttp://loca lhost:8080 /mockorcid /mock/ orcid. authorized ApiBaseUrl = http://l ocalhost: 8080 /mockorcid /mock/ orcid. oauthAutho rizeUrl = http:/ localhost :8080 /mockorcid /mock /oauth /authorize orcid. oauthToken Url = http:/ localhost :8080 /mockorcid /mock /oauth /token </pre>	<p>Setup for the mockorcid app</p>
<pre> google. maps.key= </pre>	<p>To use the Google Maps (e.g. Map of Science), you need to have a key for Google Maps. See https://developers.google.com/maps/documentation/javascript/get-api-key When you have a key, uncomment the line below and add it here</p>

resource.altmetric.disabled	Uncomment and set this to disabled if you don't want AltMetric badges
resource.altmetric.displayto=right	Display the badge to the left or right of the title (default = right). Options: left, right
resource.altmetric.badge-type=donut	Badge type to display (default = donut) Options: See AltMetric documentation - recommended settings: donut, medium-donut
resource.altmetric.hide-no-mentions=true	Hide the badge if there are no mentions (default = true) Options: true, false
resource.altmetric.badge-popover=right	Display more details about the score when you hover over the badge (default = right) Options, right, left, up, down
resource.altmetric.badge-details=right	Display extended details alongside the badge (default = none)
homePage.geoFocusMaps=enabled	When the following flag is set to enabled, the VIVO home page displays a global map highlighting the geographical focus of foaf:person individuals. See Home page customizations
multiViews.profilePageTypes=enabled	VIVO supports the simultaneous use of a full foaf:Person profile page view and a "quick" page view that emphasizes the individual's webpage presence. Implementing this feature requires an installation to develop a web service that captures images of web pages or to use an existing service outside of VIVO. See Multiple profile types for foaf:Person
http.createCacheHeaders = true	Tell VIVO to generate HTTP headers on its responses to facilitate caching the profile pages that it creates. See Use HTTP caching to improve performance Developers will likely want to leave caching disabled, since a change to a Freemarker template or to a Java class would not cause the page to be considered stale.
harvester.location = /usr/local/vivo/harvester/	Absolute path on the server of the Harvester root directory. You must include the final slash. Setting a value for harvester.location indicates that the Harvester is installed at this path. This will enable the Harvester functions in the Ingest Tools page.
visualization.topLevelOrg = http://vivo.mydomain.edu/individual/topLevelOrgURI	The temporal graph visualization is used to compare different organizations/people within an organization on parameters like number of publications or grants. By default, the app will attempt to make its best guess at the top level organization in your instance. If you're unhappy with this selection, uncomment out the property below and set it to the URI of the organization individual you want to identify as the top level organization. It will be used as the default whenever the temporal graph visualization is rendered without being passed an explicit org. For example, to use "Ponce School of Medicine" as the top organization: visualization.topLevelOrg = http://vivo.psm.edu/individual/n2862
visualization.temporal = enabled	The temporal graph visualization can require extensive machine resources. This can have a particularly noticeable impact on memory usage if The organization tree is deep, The number of grants and publications is large. VIVO 1.3 release mitigates this problem by the way of a caching mechanism hence we can safely set this to be enabled by default.

<pre> proxy. eligibleTypeList = http://xmlns.com/foaf/0.1/Person, http://xmlns.com/foaf/0.1/Organization </pre>	<p>Types of individual for which we can create proxy editors. If this is omitted, defaults to http://www.w3.org/2002/07/owl#Thing</p>
<pre> Vitro.reconcile.defaultTypeList = http://vivoweb.org/ontology/core#Role, core:Role; http://vivoweb.org/ontology/core#AcademicDegree, core:AcademicDegree; http://purl.org/NET/c4dm/event.owl#Event, event:Event; http://vivoweb.org/ontology/core#Location, core:Location; http://xmlns.com/foaf/0.1/Organization, foaf:Organization; http://xmlns.com/foaf/0.1/Person, foaf:Person; http://purl.obolibrary.org/obo/IAO_000030, obo:IAO_000030 </pre>	<p>Default type(s) for Google Refine Reconciliation Service. The format for this property is id, name; id1, name1; id2, name2 etc. For more information, see Service Metadata from this page: https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API</p>