

Using a different data store

- [Different database type](#)
- [Optional external triple store](#)

VIVO can work with databases like Oracle or SQL Server, or with external triple stores.

Different database type

To use a database other than MySQL, set these values in runtime.properties. Depending on your database, you might also need to change the basic properties relating to VitroConnection.

Property name	<code>VitroConnection.DataSource.dbtype</code>
Description	Change the dbtype setting to use a database other than MySQL. Otherwise, leave this value unchanged. Possible values are DB2, derby, HSQLDB, H2, MySQL, Oracle, PostgreSQL, and SQLServer. Refer to http://openjena.org/wiki/SDB/Databases_Supported for additional information. Earlier releases of VIVO used Jena version 2.6.4, which does not work correctly with Microsoft SQL Server. VIVO release 1.7 uses Jena version 2.10.1, which should fix this problem.
Default value	MySQL
Example value	Oracle

Property name	<code>VitroConnection.DataSource.driver</code>
Description	Specify a driver class name to use a database other than MySQL. Otherwise, leave this value unchanged. This JAR file for this driver must be added to the the webapp/lib directory within the vitro.core.dir specified above.
Default value	<code>com.mysql.jdbc.Driver</code>
Example value	<code>com.mysql.jdbc.Driver</code>

Property name	<code>VitroConnection.DataSource.validationQuery</code>
Description	Change the validation query used to test database connections only if necessary to use a database other than MySQL. Otherwise, leave this value unchanged.
Default value	SELECT 1
Example value	SELECT 1

Optional external triple store

VIVO can be configured to use a different triple store for the bulk of its semantic data, so long as this triple store supports Web-based use of the SPARQL language to query and modify its data. If you elect to use a separate triple store, note that VIVO's MySQL database is still required for basic configuration and user account data. In order to connect VIVO to an external triple store, you will need to know two URIs: the store's endpoint URI for issuing SPARQL queries that read data, and its URI for issuing SPARQL UPDATE commands. These URIs are typically kept separate in order to make it easier to secure the triple store against unauthorized edits. With Sesame, for example, the update URI is usually the query endpoint URI with `/statements` appended.

You will need to know these two URIs later when you [specify runtime properties](#).

Property name	<code>VitroConnection.DataSource.endpointURI</code>
Description	Set the endpointURI only if you wish to store semantic data in an external triple store instead of MySQL. Enter the URI of the triple store's SPARQL endpoint for querying data.
Default value	NONE
Example value	

Property name	<code>VitroConnection.DataSource.updateEndpointURI</code>
---------------	---

Description	Set the updateEndpointURI only if you wish to store semantic data in an external triple store instead of MySQL. Enter the URI at which the triple store responds to SPARQL UPDATE requests. This setting is only necessary if the triple store does not support updates via its main URI. If the endpointURI above is not set, this setting has no effect.
Default value	NONE
Example value	.