Upgrading VIVO to release 1.7

- Before Performing the Upgrade
- Noteworthy Changes
  - Upgrade of Jena libraries
  - Level 1 ORCiD integration
  - API improvements
  - Solr modularity and upgrade.
  - Improvements to data tools
  - Supported Browsers
- Upgrade Instructions
  - Download the new distribution file and unpack it into a new source directory.
  - Apply any previous changes you have made to the new source directory.
  - Apply any previous changes you have made to the RDF initialization files.
  - Run the build script
  - Run the RDB migration tool
  - Start VIVO
  - Rebuild the search index
  - Review the knowledge base migration logs.
- RDB Migration
  - A typical migration
  - Variations on migration
    - The TDB directory already exists
    - The RDB data has already been migrated
    - There are additional RDB models in the database
    - No RDB models in the database
- Knowledge Base Migration
- Review the VIVO Terms of Use
- Next Steps

Upgrading from Release 1.6.x to Release 1.7

This document contains instructions on how to upgrade your installation of VIVO from Version 1.6 (or 1.6.1, or 1.6.2) to Version 1.7. This and other documentation can be found on the support page at VIVOweb.org

If you need to do a fresh install, please consult the VIVO Installation Instructions, found on vivoweb.org, or the VIVO Installation Instructions.pdf file located in the `doc` directory of the VIVO source code distribution. The installation document also has a list of the required software and versions.

For a description of the release contents see the Release announcement for V1.7.

Before Performing the Upgrade

Create backups of:

- The VIVO distribution directory (which contains the source for VIVO 1.6 (or VIVO 1.6.1, or 1.6.2))
- The VIVO home directory (pointed to by your build.properties file)
- The webapps directory in Tomcat
- MySQL database (most people use mysqldump to create the backup)

If you have used temporary models in the database to stage ingested data, you will want to clear out any unneeded models that remain listed on the `Manage Jena Models` page (under `Ingest tools`). This step is especially important if these temporary models contain blank nodes, as this may cause unwanted or duplicate data to appear following the upgrade.The upgrade process is similar to the initial install process with the following exceptions:

- You do not need to reinstall MySQL or recreate the MySQL database. Please backup your MySQL database as noted above.
- The root account will keep the password that was previously set on it. It will not return to the default password. Any user accounts that you have created will also be preserved.

The mechanism for storing "configuration" data has changed in VIVO 1.7. You must run the RDB migration program to accomplish this change. See RDB Migration below.

Noteworthy Changes

Upgrade of Jena libraries

The Jena libraries used by VIVO have been upgrade from Jena 2.6.4 to Jena 2.10.1. This upgrade helps VIVO remain compatible with a variety of tools. Initial testing indicates that the new libraries are more efficient, resulting in a 15% improvement in load times for large profile pages.

The newer Jena libraries no longer support the Jena RDB implementation, which VIVO used to store configuration data, such as display options and user accounts. In VIVO 1.7, the configuration data has been converted to the Jena TDB implementation, which is favored. Sites that upgrade from a previous VIVO are required to run a migration utility program before running VIVO 1.7

Level 1 ORCiD integration

VIVO 1.7 provides the option of simple integration with the ORCiD registry of research IDs. If the ORCiD integration is enabled, researchers may add their ORCiD ID to their VIVO profile page, and may also choose to add a link from their ORCiD page to their VIVO profile.

VIVO-ORCiD integration is available only to ORCiD member institutions. In order to enable the integration, the VIVO site must be registered with ORCiD as an approved application.

API improvements

A new API has been added to support SPARQL queries. This API is similar in form to the SPARQL update API introduced in VIVO 1.6, supporting authentication per request, and providing results in a variety of formats, by content negotiation.

The "ListRDF" API call has been improved to support content negotiation. Also, a bug which limited the size of the exported lists was fixed.

Solr modularity and upgrade.

Some VIVO sites prefer to use a search engine other than Solr. In VIVO 1.7, the relationship between VIVO and Solr has been summarized in a clean interface. VIVO sites should find it much easier to substitute the search engine of their choice. The core VIVO development team is eager to partner with such sites to create search engine adapters which can be used by the VIVO community.

Also in VIVO 1.7, the Solr search engine in VIVO has been upgraded from Solr 3.1.0 to Solr 4.7.2, for increased speed and reliability.

Improvements to data tools

A new tool has been added to the Site Administrator pages, allowing the administrator to export the entire RDF contents of VIVO. Because the data is exported as "quads", the graph information is preserved. This export capability can facilitate analysis of VIVO data, or transfer of data from one VIVO installation to another.

Supported Browsers

For this release, the following browsers are supported.

- Mac:
  - Chrome 30.0.1599.69 and above
  - FireFox 3.6.28, 10.0.12, 24, 30
  - Opera 12.15
  - Safari 5.0.3 and above
- PC:
  - Chrome 25.1364.2 and above
  - FireFox 10.0.12, 24, 30
  - Internet Explorer 8, 9, 10
  - Opera 12.02

Upgrade Instructions

Download the new distribution file and unpack it into a new source directory.

Apply any previous changes you have made to the new source directory.

> ***Special notes regarding source files***
>
> - *This process assumes any changes made to the application were made in the source directory and deployed, and were not made directly within the Tomcat webapps directory.*
> - *In many cases, simply copying the modified files from your original source directory will not work since the files on which they are based have changed. It will be necessary to inspect the new source files and add any changes to them at that time.*
> - *NIH-funded VIVO implementations will need to apply the Google Analytics Tracking Code (GATC) to `googleAnalytics.ftl` in the theme:*
>
>   *`[new_source_directory]/themes/[theme_dir]/templates/googleAnalytics.ftl`*
>
>   *A sample `googleAnalytics.ftl` is included in the built-in theme. This file serves only as an example, and you must replace the tracking code shown with your institution's own tracking code. For additional information about the GATC for*

> *the NIH-funded VIVO implementation sites and a copy of your institution's tracking code, see the VIVO Google Analytics wiki page.*

Apply any previous changes you have made to the RDF initialization files.

Run the build script
Stop Apache Tomcat and from your VIVO source directory, run ant by typing: `ant all`

Run the RDB migration tool

The Jena libraries have been upgraded, and VIVO can no longer read the database tables that hold your user accounts and display options.

From your VIVO source directory, run the RDB migration tool by typing: `ant migrateConfigurationModels`

For more details, see RDB Migration, below.

Start VIVO
Start Apache Tomcat and log into VIVO as the root user when the upgrade is completed. Depending on the size of your database, the migration process may take up to several hours. When it is complete, you will see a message in the catalina.log file that the server has started.

```
INFO: Server startup in XXXXX ms
```

Rebuild the search index
As root or an administrator, request a rebuild of the Solr search index: Go to the "Site Admin" page and click on "Rebuild Search Index" under the heading "Site Maintenance".

Review the knowledge base migration logs.

This is usually an important step in VIVO upgrades. However, the upgrade from VIVO 1.6 to VIVO 1.7 does not require a knowledge base migration.

RDB Migration

RDB is a triple-storage mechanism that is no longer supported by the recent Jena libraries. VIVO will now store it's configuration data in TDB, which uses a file directory instead of a database.

The migration tool uses the old libraries to read the VIVO configuration data from RDB. It creates a new subdirectory called `tdbModels` in your VIVO home directory, and stores the configuration data there.

A bug has been found in the script that runs the RDB Migration tool. On some systems, the tool does not respond to user input, so when you are prompted with

```
Continue? (y/n)
```

s simple, and is detailed in the JIRA issue: Unable to locate Jira server for this macro. It may be due to Application Link configuration.

A typical migration

A typical migration session might look like this:

```
ant migrateConfigurationModels
Buildfile: /Users/jeb228/git/VIVO/build.xml

migrateConfigurationModels:

clean:

getBuildProperties:

getRuntimeProperties:

setup:
    [mkdir] Created dir: /Users/jeb228/git/Vitro/utilities/rdbmigration/.work

compile:
    [javac] Compiling 1 source file to /Users/jeb228/git/Vitro/utilities/rdbmigration/.work

run:
     [java] Migrating 645 triples in 4 models to TDB files in '/Users/jeb228/Development/Performance/Weill
/vivoHome/tdbModels'
     [java]    Existing triples will be over-written.
     [java] Continue? (y/n)
y
     [java]  WARN [main] (SetupTDB.java:755) - No BGP optimizer
     [java]   copied  220 triples from http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadata
     [java]   copied   76 triples from http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadataTBOX
     [java]   copied    6 triples from http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadata-
displayModel
     [java]   copied  343 triples from http://vitro.mannlib.cornell.edu/default/vitro-kb-userAccounts

all:

BUILD SUCCESSFUL
Total time: 11 seconds
```

When you invoke `ant migrateConfigureionModels`, ant compiles the RDB migration utility and runs it.

The RDB migration utility:

- Reads your `build.properties` file, to locate your VIVO home directory.
- From your home directory, reads your `runtime.properties` file to find your database parameters: username, password, and JDBC URL.
- Calculates the amount of configuration data to be migrated.
- Asks for your approval to continue.
- Copies the data.
- Adds a table to your database, recording that the migration has happened, and when.

Variations on migration

The TDB directory already exists

If you already have a `tdbModels` sub-directory in your VIVO home directory, and it is not empty, the migration utility will warn you. You may choose to continue, but the contents of the `tdbmodels` directory will be replaced.

```
     [java] A directory of TDB files exists at '/Users/jeb228/Development/Performance/Weill/vivoHome/tdbModels'.
     [java]    Migration will replace the existing triples.
     [java] Continue? (y/n)
y
```

The RDB data has already been migrated

If your database already contains a table named `vivo_rdb_migrated`, the migration utility will warn you. You may choose to continue, if for some reason the migration must be repeated.

```
     [java] It looks like this RDB data has already been migrated to TDB, on 2014-06-18
     [java]    (found a table named 'vivo_rdb_migrated')
     [java] Migrate again? (y/n)
y
```

There are additional RDB models in the database

The VIVO ingest tools allow you to create additional models and store them as RDB in the database. If these models are large enough, they can cause the RDB migration utility to run out of memory and abort. The RDB migration utility can use up to 2 gigabytes of storage, so is should be capable of copying tens of thousands of triples.

In this case, the utility will provide you with the option of migrating all of the models, or only the four models that VIVO requires.

```
     [java] VIVO requires only these models from RDB:
     [java]    http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadata
     [java]    http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadata-displayModel
     [java]    http://vitro.mannlib.cornell.edu/default/vitro-kb-displayMetadataTBOX
     [java]    http://vitro.mannlib.cornell.edu/default/vitro-kb-userAccounts
     [java] Your RDB triple-store contains these additional models:
     [java]    http://my.extra.model
     [java] Migrate all models, or only the required models?
     [java]    Enter 'all' or 'only', or anything else to quit.
all
     [java] Migrating 65402 triples in 5 models to TDB files in '/Users/jeb228/Development/Performance/Weill
/vivoHome/tdbModels'
     [java]    Existing triples will be over-written.
     [java] Continue? (y/n)
y
```

No RDB models in the database

If you have started with an empty database, no migration is needed, nor is it possible:

```
run:
     [java] -------------------------------------------------------------
     [java] The database at 'jdbc:mysql://localhost/vivo' contains no RDB tables.
     [java] -------------------------------------------------------------

all:

BUILD SUCCESSFUL
Total time: 1 second
```

Knowledge Base Migration

This is usually an important step in VIVO upgrades. However, the upgrade from VIVO 1.6 to VIVO 1.7 does not require a knowledge base migration.

Review the VIVO Terms of Use

VIVO comes with a "Terms of Use" statement linked from the footer. The "Site Name" you assign in the "Site Information" form under the **Site Admin** area will be inserted into the "Terms of Use" statement. If you want to edit the text content more than just the "Site Name", the file can be found here:

`[vivo_source_dir]/vitro-core/webapp/web/templates/freemarker/body/termsOfUse.ftl`

Be sure to make the changes in your source files and deploy them to your tomcat so you don't lose your changes next time you deploy for another reason.

Next Steps

Now that you have VIVO up and running, please refer to the Site Administrator's Guide for information about its operation.