Upgrading VIVO

- Overview
- Changes to runtime.properties
  - Location
  - Password hashing settings
- Upgrading The Triple Stores

Overview

Upgrading to VIVO 1.10 requires upgrading your triple stores (content and configuration). A procedure is provided below. You should consider:

1. System downtime is required to upgrade the triple stores
2. SPARQL queries will need to be checked for use of string datatypes. See Data types for string and language
3. Applications directly accessing the SDB triple store will need to be upgraded to use Jena 3 libraries.
4. With the upgrade to Jena 3.x, Java 8 is required. The Maven projects have been upgraded to state a dependency on version 8, and Maven will not run without it.
5. If you have customizations, please see
   a. Preserving Customizations During Build for processes to include your customizations in a VIVO build
   b. Additional Considerations for notes which may impact your customizations

Changes to runtime.properties

Location

For VIVO 1.10, the preferred location of `runtime.properties` has changed from the `<vivo_home>` directory to `<vivo_home>/config`. VIVO will raise a warning on startup if runtime.properties is found in `<vivo_home>` or both `<vivo_home>` and `<vivo_home>/config`. Continue startup by refreshing or clicking continue. Move `runtime.properties` into the config directory to avoid the warning. Note, jenatools expects to find runtime.properties in <vivo_home>.

Password hashing settings

VIVO 1.10 includes security enhancements to the way passwords are stored in VIVO. Three new settings are now required in runtime.properties. Add the following to an older version of runtime.settings (default settings shown):

runtime.settings (snippet)

```
argon2.parallelism = 1
argon2.memory = 1024
argon2.time = 1000
```

For a full explanation of the new settings, see example.runtime.properties.

Upgrading The Triple Stores

Upgrading the triple stores (there are two - content and configuration) involves dumping the contents of your stores, and then reloading them, using tools provided with VIVO.

In order to upgrade your triple stores, use the following steps (replace `<your-settings.xml>` and `<vivo_home>` with the appropriate values for your system.

1. Stop Tomcat - it is vital that Tomcat, and any other applications that may access the triple stores, are not running during this process.
2. Run `mvn clean install -s <your-settings.xml>` in your VIVO 1.10.0 area to update your web application and home directory
   This will install the required tools into your `<vivo_home>/bin` directory.
3. To export your triple stores, use the jena2tools utility provided with VIVO 1.10.0, in `<vivo home dir>/bin`, specifying the export command, as shown below.

   ```
   java -jar jena2tools.jar -e -d <vivo_home>
   ```

   Arguments:

-d - the location of the Vitro/VIVO home directory

-e - run in export mode

On execution, the program will read your configuration files, find your VIVO configuration within the vivo home directory, and get the necessary information to connect to your configuration triple store, and your content triple store. If your triple store(s) are not SDB or TDB backed, they will be skipped.

jena2tools will then extract the contents of the triple stores, and write them to `<vivo_home>/dumps`

In rare cases `jena2tools` will fail with a `java.lang.NullPointerException`. This can occur if VIVO was not properly shut down before upgrading.

The preferred fix for this is to restore VIVO 1.9.x and see that it is properly started and shut down before upgrading. If this is not practical, another workaround is to delete the file named `tdbModels/journal.jrnl` in the Vitro/VIVO home directory. This may result in the loss of the most recent login information.

4. Check that the export has completed - you should have a `<vivo_home>/dumps` directory, that contains the files `configuration.trig` and `content.trig`.

5. Empty your triple stores
   - Drop your database and recreate it as empty, just as you would for creating a new VIVO install. jena3tools must find an empty database (no tables) as named in your `runtime.properties` and will recreate your content triple store using the triples produced by jena2tools

     ```
     mysql> DROP DATABASE vitrodb;
     mysql> CREATE DATABASE vitrodb CHARACTER SET utf8;
     mysql> GRANT ALL ON vitrodb.* TO 'vitrodbUsername'@'localhost' IDENTIFIED BY 'vitrodbPassword';
     ```

   - Delete all files in `<vivo_home>/tdbModels`. Jena3tools will rebuild your configuration tdbModels based on the content created by jena2tools

     ```
     rm -rf <vivo_home>/tdbModels
     ```

6. Reload your triple stores

   Having exported your triple stores, you can reload them using jena3tools, also available with VIVO 1.10, specifying the import command.

   ```
   java -jar jena3tools.jar -i -d <vivo_home>
   ```

   Arguments:

   -d - the location of the Vitro/VIVO home directory

   -i - run in import mode

   On execution, the program will find your VIVO configuration within the home directory, as well as the dumps that you have created with jena2tools. It will import them into the SDB and TDB triple stores, based on the configuration of your VIVO instance.

   jena3tools will be present in <vivo home dir>/bin when you install the 1.10.

   Note that this can take a while. A rough guide is to expect about 600 triples per second to reload. (Roughly 1 hour per 2 million triples).

7. Restart Tomcat

   VIVO checks its filegraphs when starting. Restarting for the first time after an upgrade will take some time.