# 20110114 UF Harvester Training

This was a meeting between the UF developers and UF implementors. This meeting served as training to the local team, and development of documentation for the harvester.

**Date:** 1/14/2010

**Attendees:** Alex Rockwell, Christopher Case, Christopher Haines, James Pence, Michael Barbieri, Nicholas Rejack, Nicholas Skaggs, Stephen Williams

## Agenda

- Development Toolset
  - Eclipse Setup
  - Maven setup
  - SVN checkout
  - Project settings
    - Project Layout
    - Packages and Classes
    - Configuration and Script Files
- Harvester Overview
  - 1.0 Modified Features and affect on scripts
  - Java Optimizations
- PubMed, PeopleSoft, & DSR Harvests
  - Fetch Configuration
  - XSLT Overview and Modification Overview
  - Script Flow
    - What the script is doing (flags explained, etc)
    - How to backup in stages and restore from backups
    - Logging provided and what to look for

## Notes

under tasks

- driver, connection, username and password need to be set.
- The xml also includes which tables are being fetched from.

the project so far is maintained only Linux, though Windows is possible. MACOS

Each module can be ran as a java library or within command line scripts

src/main/java contains the various modules of the harvester

- Score - calculates the scores and puts them in a separate model
  - When calling on the algorithms they are assigned weights
- Algorithm - contains the various scoring mechanisms for equality
  - Equality Test - for just 100% matches.
- Match - matches with the score data.

- ChangeNamespace - takes all of one namespace and changes them to a new namespace. Suggested for use after scoring and matches, cleans up the remainder
- Jena is a RDF data store.
- JenaConnect – (rdb,sdb,mem versions) an interface to use/expose the jena models. SPARQL querys execution. Outputs to the command line
- Qualify - allows modification of the data. May develop a new way of approaching
- Diff – creates the differences of the jena models using the jena methods.
- Transfer – from Record Handler or model to a jena model (not from a model to record handler)
- the args folder – for dealing with the command line arguments.
- Merge – specify a regex style call to merge records into a single record. Allows cross table references during translation. (ie. Fetch  merge translate)
- XSLTranslator – uses the Java library to translate the data
  - the other translate options need work.
- config/datamaps contains the various XSL files for the data transforms.
  - Including regex abilites, separated out into different Templates.

src/main.resources

- LogBack – the logging system which is configuarble in “logback.xml” (Fixing the Timezone? GMT right now.)

- XSLT templates
  - creating links with one template and creating a stub for the uri.
  - The namespaces used within the harvest is specific to the type of object fetched.
  - They use their own regex formats. (Be sure to reference xsl version

- Pubmed – Parsing out the name is happening using the analyze string

    - MESH heading - descriptor and qualify skipped.
    - Types are forced to lowercase for identification. (fixes Types)

Commit standard steps:

- 1.update to head
- 2.inspect or resolve any merges.
- 3.Commit the changes.

- Harvester steps:
    - 1.Fetch
        - (a)the easiest way to inspect the data is a to use a text file record handler
            - i.a non text file record handler suggested for production.
    - 2.Merge
        - (a)used in people soft to facilitate data relations
        - (b)Not needed for JDBCFetched data
    - 3.Translate
        - (a)turns data into rdf
    - 4.Transfer
        - (a)moves data to a Jena model
    - 5.(Data can now me mounted and viewed in VIVO but is not suggested)
        - (a)not to be used in production

- JenaModels
    - Added a type parameter for defining the db format of the back-end jena db
    - The default namespace is found here to ensure consistency.

- Drivers
    - H2, HSQLDB,MYSQL are in the dependencies.

- Record Handlers
    - XML is the version for raw XML
    - RDF is the RDF/XML version

Scripts
the scripts env file sets many of the used parameters

- Date format, vivo connection,the various tools locations and optimization flags.

In the PubMed the the algorithms for the work email and name parts are implemented.
In the DSR

Match needs:

- -r to replace the found matches
- -lpred1=pred2 to form a link
- -o returns an output model of the matches.
- -t is the the threshold within which the matches will be considered true.