VIVO Harvester User Guide 1.0

Installation

Please Note:

1. These instructions assume that you are performing a clean install. Product functionality may not be as expected if you install over an existing installation of an earlier version.
2. VIVO Developers: These instructions are for installing, configuring and running the VIVO harvester. Instructions for developing on the VIVO Harvester library can be found at Harvester Development Toolkit.

The VIVO Harvester is a library of tools written in Java for exporting data from external systems and importing it into VIVO. The code is released in two ways, as a debian file and as tar file. Linux machines that use apt-get style repositories (debian, ubuntu) can install the debian file by downloading it and using dpkg -i filename.deb to install. The tarball can be unpacked on any OS where java is supported and run.

Walk-through Videos

In addition to the wiki information, screencasts of example harvester runs using the 1.0 version of the harvester can be found here

Requirements

- Required Software
  - Sun Java 6
  - Maven (maven2 or maven3): This is required to compile the code. It automatically handles the retrieval of any third-party dependencies, performs code compilation, and performs unit tests.
  - Subversion: This is required for code checkout
- Deb Version
  *__The debian package installs itself into the /usr/share/vivo/harvester/__ directory. It should prompt before updating config files when installing a new version of the harvester over an old version.
- Tarball
  - Unpacks into any directory you wish and can be run with no additional installation. This is a great option if you don't have root access on the machine you're working on.

Execution

For an overview of the components used and how they work together in a harvest, see Typical harvest

There are two ways to run the VIVO Harvester either as command-line tools (useful in a bash script) or use it as a Java library. We'll cover both ways.

Command-Line

All the Harvester tools, when used via the command-line share some common parameters and parameter formats. General information about the Harvester tools on the command-line can be found at Harvester Tools.

Bash Scripts

DEPRECATED – the struckout text is only for harvester 1.1.1 and previous versions

The first thing you should do is open your Environment Config File and make sure everything is in order. This file is located in scripts/env. Next you should look at your vivo configuration file to make sure this points to your VIVO installation. This file is located in config/models/vivo.xml. You can check out our example scripts or our walk throughs for how to create a bash file using the environment file.

Starting with the Harvester 1.2, there are several example scripts that now ship in the example-scripts folder. Following is a walkthru for each of them:

- National Data Source Examples
  - PubMed
  - MODS
- Local Data Source Examples
  - JDBC
    - PeopleSoft(UF)
    - DSR(UF)
  - D2RMap
  - CSV

Other things

- RecordHandlers
  *Backup segments of the harvest process

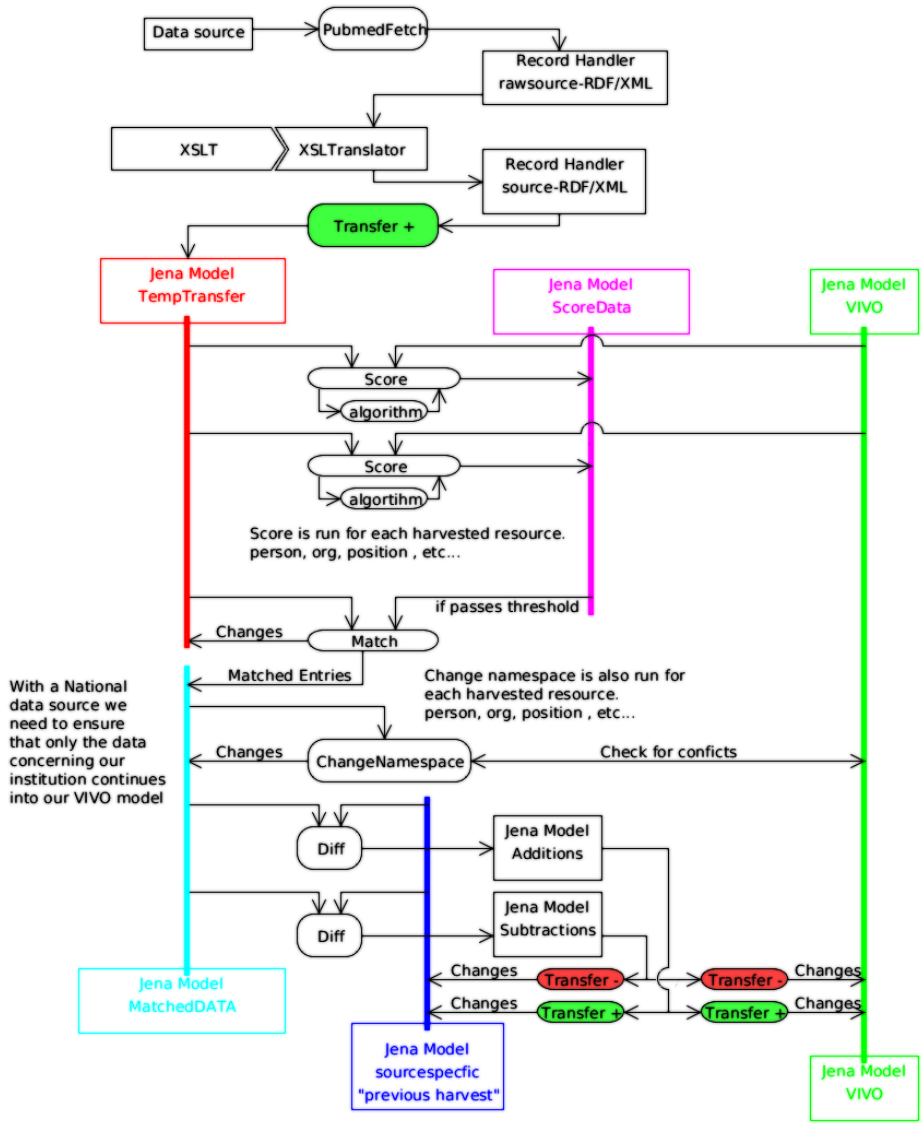
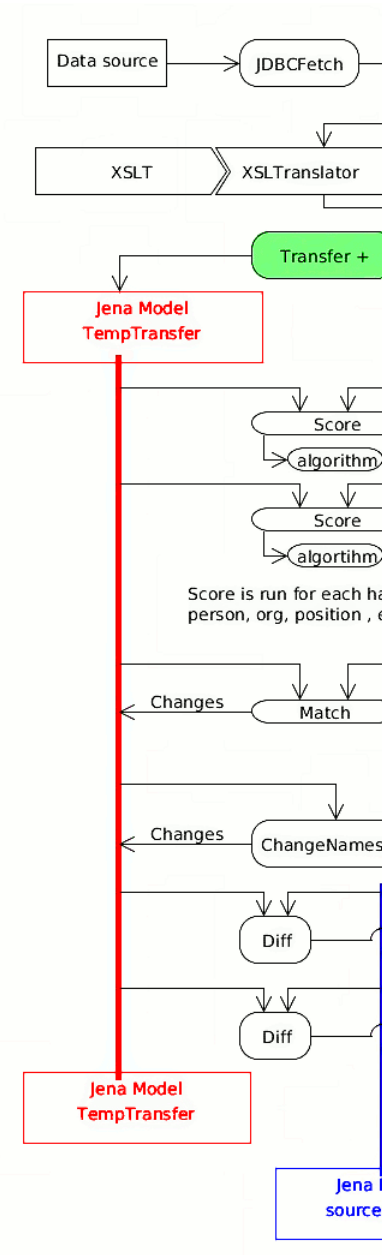
- Temporary Models as used by the harvester
  *XPathTool use in the Environment Config File

Library

Using the harvester as a library works well when you want to embed it's functionality into an already existing java application. Utilizing the tools are straightforward. See our Javadoc [1] documentation for specifics on each tool. A working example harvest is also shipped starting with the harvester 1.0 release. See the DemoPSMerge.java file included the demos folder.

Data Workflow

These images are diagrams to show how the data flows in some example cases.

National data	Local data
	

1. External Data Source - This is the foreign source
2. Fetch - Retrieves data from foreign source
3. Raw Data - A simple database or as simple XML
4. Translate - Turns the raw data into RDF
5. RDF - RDF models which can be dumped into RDF/XML.

6. Score] - First find similarities and rate them, second determine and apply matches based on a threshold of difference.
7. Qualify - Changes any unmatched data
8. Transfer (Update) - move into a vivo model (through an update process if possible.)
9. Vivo - Final model in RDF visible from the webapp. |

PubmedFetch	JDBCFetch	Fetch

Tools

The harvester is a collection of tools. Each of the following pages has detailed information about how to use that tool, what parameters it takes, and what methods it exposes. Please refer to these pages for more information on how to use each of the tools for the VIVO Harvester.

- Fetch
  - OAIFetch
  - NIHFetch
    - PubmedFetch
    - NLMJournalFetch
  - RDB Harvesters
    - JDBCFetch
    - D2RMapFetch
- Translate
  - GlozeTranslator
  - SPARQLTranslator
  - XSLTranslator
- Score/Match
  - Score
  - Algorithm
  - Match
- Transfer
- Diff
- Qualify
  - ChangeNamespace
  - RenameResources
  - Smush
- Utilities
  - JenaConnect
  - RecordHandler
  - ArgParser
  - Merge
  - DatabaseClone
  - XPathTool
  - CSVtoJDBC

Upgrading

Currently the VIVO Harvester does not have an upgrade path, as all previous versions were considered beta versions. With the 1.0 release you can expect all subsequent releases follow standard software upgrade guidelines.