

Peoplesoft Example Script 1.2

Your first Harvest

- Change directory to example-scripts/example-peoplesoft
- Edit the vivo.model.xml file
 - Set the dbURL, dbUser, dbPass, and Namespace
 - For more information on these parameters and their use, please see [Harvester vivo configuration file](#)
- Edit changenamespace-people.config.xml, changenamespace-departments.config.xml, and changenamespace-positions.config.xml files and set the namespace parameters in each one to be your vivo namespace
 - For more information on these parameters and their use, please see [ChangeNamespace](#)
- Edit the run-peoplesoft.sh file and set the HARVESTER_INSTALL_DIR= to be the directory you unpacked the harvester in
- Run bash run-peoplesoft.sh

The first run

Three folders will be created

- logs
- data
- previous-harvest

The logs folder contains the log from the run, the data folder contains the data from each run, and the previous-harvest folder contains the old harvest data for use during the update process at the end of the script. While your testing, I would recommend treating each run as the first run (so no update logic will occur). You can do this by removing the previous-harvest folder before running again.

Inside the data folder, you will find the raw records utilized during the ingest. To see what rdf statements went into VIVO, you can view the vivo-additions.rdf.xml file. Conversely, to view what the harvester removed (because of updated data), you can view the vivo-subtractions.rdf.xml file. This file will be blank on your first run, since you have no previous harvest to compare the incoming data against.

Optimizing

Once your ready to run a large dataset, it is advisable to the record storage from files to a database. Although this will make it harder to find individual records, speed and performance will be increased during the fetch and translate stage. To do so:

- Edit the nano raw-records.config.xml to use TDB, which is a semantic data store

```
<RecordHandler>
  <Param name="rhClass">org.vivoweb.harvester.util.repo.JenaRecordHandler</Param>
  <Param name="type">tdb</Param>
  <Param name="dbDir">data/raw-records</Param>
</RecordHandler>
```

- Edit the translated-records.config.xml to use TDB, which is a semantic data store

```
<RecordHandler>
  <Param name="rhClass">org.vivoweb.harvester.util.repo.JenaRecordHandler</Param>
  <Param name="type">tdb</Param>
  <Param name="dbDir">data/translated-records</Param>
</RecordHandler>
```