ARCOS CLI Tutorial

Full instructions available online: https://github.com/ge-high-assurance/RACK/blob/master/cli/README.md

How to get started

We use a virtual environment to keep all the Python dependencies contained.

```
$ cd RACK/cli
$ virtualenv venv
$ . venv/bin/activate
(venv) $ pip install -r requirements.txt
(venv) $ pip install .
```

Usage help is available within the command an online in the README

```
(venv) $ rack --help
(venv) $ rack data --help
(venv) $ rack data import --help
```

Things the CLI can do for you

- Update the data model
- Store and retrieve nodegroups
- Store and retrieve data

How we use CLI to configure RACK

The CLI can fully populate a RACK instance with an ontology, nodegroups, and data.

We use a script during development to automate deployment of changes.

```
# setup-rack.sh

# Populate the data model
rack model import --clear ../RACK-Ontology/OwlModels/import.yaml
rack model import ../Turnstile-Ontology/99-Utils/import.yaml
rack model import ../GrammaTech-Ontology/import.yaml
rack model import ../STR-Ontology/import.yaml
```

```
# Populate the nodegroups
rack nodegroups delete --yes --regexp --ignore-nonexistent "^ingest[0-9]+" "
rack nodegroups import ../Turnstile-Ontology/99-Utils/NodeGroups
rack nodegroups import ../nodegroups/queries

# Populate sample data
rack data import --clear ../RACK-Ontology/OwlModels/ARP-4754A.yaml
rack data import --clear ../RACK-Ontology/OwlModels/DO-330.yaml
rack data import --clear ../RACK-Ontology/OwlModels/DO-178C.yaml
rack data import --clear ../RACK-Ontology/OwlModels/MIL-STD-881D.yaml
rack data import --clear ../Turnstile-Ontology/99-Utils/Data/Model.yaml
```

CLI to SemTK connection

Import and export *nodegroups*

Inside the RACK web interface we can Load, Save, and Delete nodegroups through the *Nodegroup* menu item. These steps can be automated with the CLI.

See available nodegroups

```
$ rack nodegroups list
```

Delete unneeded node groups

```
$ rack nodegroups delete --regexp "^query "
$ rack nodegroups delete-all
```

Export nodegroups to disk

```
$ rack nodegroups export "^query " path/to/directory
```

Import nodegroups from disk

```
$ rack nodegroups import path/to/directory
```

Data Ingestion

Using the RACK web interface we can ingest data against a nodegroup using the *Import* tab.

Using the CLI we can automate the process of importing many CSV files against many nodegroups.

Ingestion steps are captured in a YAML file:

```
data-graph: "http://rack001/data"
ingestion-steps:
- {csv: "SourceCodeTracing.csv", nodegroup: "Ingest-SourceCodeTracing"}
- {csv: "SourceCode.csv", nodegroup: "Ingest-SourceCode"}
- {owl: "DirectImport.owl"}
- {ttl: "RawData.ttl"}
```

Which can be provided to RACK:

```
$ rack data import path/to/import.yaml
```

Data Queries

We can automate the process of loading a nodegroup in RACK

Queries can be run in human-readable output

```
$ rack data export Ingest-HighLevelRequirements http://rack001/data
```

or CSV generation mode

```
$ rack data export --format csv Ingest-HighLevelRequirements http://rack001/
```

For sanity checking expected outputs we can just check number of results

```
rack data count Ingest-HighLevelRequirements http://rack001/data
```

Updating ontology

In RACK the model can be updated using the *Import* tab's *Graph Tools*. With the CLI we can automate this process.

Ontology ingestion steps are stored in YAML format

```
files:
- AGENTS.owl
```

```
  - ANALYSIS.owl
  - CONFIDENCE.owl
  - DOCUMENT.owl
```

We can process these steps with the CLI

```
$ rack model import ../RACK-Ontology/OwlModels/import.yaml
```

Examples in the RACK Repository

Nodegroups

- `nodegroups/ingestion`
- `nodegroups/queries`
- `Turnstile-Ontology/99-Utils/NodeGroups`

Data files

- `Turnstile-Ontology/99-Utils/Data`