Daexim

Welcome to Daexim

Introduction

Data Export/Import or daexim (pronounced 'deck-sim') for short.

Data Export/Import provides RPCs to request the bulk transfer of ODL system data between data stores and the local file system. The target use cases concern system upgrade, enabling the development of administrative procedures that make reconfigurations of the base system without concern of internal data loss. One specific use case concerns changing a YANG model and its stored data against the practices documented in RFC 6020 Section 10 "Updating a Module". Though upgrade to the new model may be exactly the desired result, it is not reasonable to ask ODL to internally cater for multiple revisions of the same module with deliberate version incompatibilities. The proposed mechanism allows such a model change to be considered and made.

Each file is specified to contain JSON-encoded data with a schema that is version-independent from ODL. As such, file data may be readily produced by, consumed by and transferred between different versions of ODL. Also, these files may be easily processed and modified using external tools.

Data Export produces a models declaration file and one or more data files. The models declaration file records exactly which models were loaded (by module name, revision date and namespace.) The data file (s) contain data store data as per draft-ietf-netmod-yang-json.

Data Import may take a models declaration file and zero or more data files. The models declaration file is used to check that the listed models are loaded before importing any data. Data is imported into each data store in turn with one transaction executed for each data store, irrespective of the number of files for that data store, as inter-module data dependencies may exist. Existing data store data may be cleared before importing.

Data Import is intended to be performed at, or very soon after, system boot/start time to support upgrade scenarios and is not intended as a general purpose alternative to RESTCONF.

This feature leverages the existing infrastructure provided by MD-SAL and yangtools.

Data for export is read from the data store DOM using the controller's binding independent APIs. One transaction is performed per data store. The models declaration is exported at the same time and is generated from the modules listed in the global schema context. The export is performed using the yangtools JSON codecs, benefiting from their special provisions to minimize memory footprint.

The export plugin is designed to operate in a clustered environment. A client is informed of the files that are produced against each node name, where the node name is read from the cluster configuration. In a cluster environment, data may not be present on all nodes and therefore exported files on different nodes may need to be reconciled. Data reconciliation is out of scope for the initial contribution and must therefore be performed offline.

When importing, modules listed in the models declaration are tested for presence in the global schema context. Data for import is written to the data store DOM using the controller's binding independent APIs. As for export, one transaction is performed per data store.

Project Facts

Project Creation Date: Oct 13, 2016

Lifecycle State: Incubation

Type: Application

Primary Contact: Shaleen Saxena <shaleen.external@gmail.com>

Project Lead: Shaleen Saxena <shaleen.external@gmail.com>

Committers:

- Ajay Lele <ajayslele@gmail.com>
- Anton Ivanov (aivanov) <anton.ivanov@kot-begemot.co.uk>
- Balaji Varadaraju (bvaradar) <bvaradar@brocade.com>
- David Spence (dspence) <david@roughsketch.co.uk>
- Michael Vorburger <vorburger@redhat.com>
- Richard Kosegi <richard.kosegi@gmail.com>
- Shaleen Saxena (ssaxena) <shaleen.external@gmail.com>
- Sreekalyan Devaraj (sdevaraj) <kalyan.ds@gmail.com>

IRC: freenode.net **#opendaylight**

Mailing List: daexim-dev@lists.opendaylight.org

Archives: mailing list archives

Meetings: See global meetings page

Repository: git clone https://git.opendaylight.org/gerrit/daexim

Jenkins: jenkins silo

Gerrit Patches: code patches/reviews

Bugs:

- open bugs

Documentation

Getting Started for Users

- User Guide

Getting Started for Developers

- Developer Guide

Release Plan

- Daexim: Nitrogen: Release Plan
- Oxygen: Release Plan

Release Notes

- Nitrogen: Release Notes
- Oxygen: Release Notes
- Fluorine: Release Notes
- Neon: Release Notes
- Sodium: Release Notes
- Magnesium Release Notes