Plastic

Welcome to

Introduction

The Plastic Project Proposal is a great place to get the five minute read of what Plastic is and why you might want to consider using it. A portion of that content is repeated below.

What is a mapping problem?

- Occurs in internals of a system behind endpoints
- ODL context – moving from northbound to southbound representations
- Sometimes need to trivially convert data representation
- JSON, XML, YML, other parse-able formats
- Sometimes need to change abstractions (1:1, N:1, N:N)
- Morph one model into completely different model
- Morph N models into one model

ODL/plastic Advantages

- Pay-as-you-go for complexity (field deployable changes)
- Declarative representations are emphasized (clear schemas)
- Translation-by-intent (say what you want, not how to do it)
- Deeper levels of abstraction to help keep custom logic schema-independence
- Can specify arbitrary morphing via plug-ins in JVM language
- Understands breaking large mapping problems up (both time and space) into small chunks

Solves problems like…

- Schema changes for device configurations across releases
- No more hard-wired dependency on vendor libraries
- In-the-field updating to support multiple versions of devices
- Light weight specifications avoid religiosity around “DRY"

Project Facts

Project Creation Date: Nov 1st, 2019

Lifecycle State: Incubation

Type: Kernel

Primary Contact: Allan Clarke <aclarke-at-pobox.com>

Project Lead: Allan Clarke <aclarke-at-pobox.com>

Committers:

- Allan Clarke <aclarke-at-pobox.com>
- Balaji Varadaraju <bvaradar@luminanetworks.com>
- Tejas Nevrekar <tnevrekar@luminanetworks.com>

IRC: freenode.net #opendaylight-plastic

Mailing List: plastic-dev@lists.opendaylight.org
Archives: mailing list archives

Meetings: See Community Meetings

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

Jenkins: jenkins silo

Gerrit Patches: code patches /reviews

Bugs:

- open bugs

Documentation, Tutorials & Examples

- Getting Started
- Docs in the repo itself or online here
- Get the latest release here (and older releases)

Documentation

- An Introduction (that is too detailed and needs rewriting)
- A Best Practices document
- A document on running the stand-alone command line version of Plastic called Plastic Runner (look for the only .tar.gz on Nexus)
- An Authoring document aimed at advanced coding usages (this should evolve into a reference over time)
- Announcement and presentation of Plastic at 2019 DDF

Tutorial

Both parts of the tutorial below include examples ready to run via Plastic Runner

- A No Coding Tutorial on how to configure translations without using any coding (for the translation itself)

- A Coding Tutorial that shows how to create simple plugins that are automatically picked up during the translation process

Source

- Git clone below for read-only access

https://git.opendaylight.org/gerrit/plastic.git

- Git clone below for full access

```
ssh://your-name-here@git.opendaylight.org:29418/plastic.git
```

Roadmap

Acknowledgments

- Allan Clarke, originator, architect, and primary implementor
- Lumina Networks, Inc, for allowing this code to be open-sourced
- Balaji Varadaraju, for discussing ideas and requirements wrangling
- Mike Arsenault, for discussing ideas and contributing
- Shaleen Saxeena, for pushing the boundaries around requirements

Open Meetings

Please contact hosts in case of any issues.

- Topic: ODL - Weekly Plastic Project meeting
- Time: TBA