Bliss

A laissez-faire SAGA Implementation

Bliss Intro | 11/02/11 | Ole Weidner

Design Goals

- Pure Python Implementation (2.4 +)

- *Pythonic* API Feeling

- Simple Installation and Deployment

- Simple Adaptor Development

- Demand-Driven v.s. Standard-Driven

- A Playground for New Ideas!

Status Quo

- GFD.90 Base Classes a.k.a. *Look & Feel*

  - Object, Url, Session, Context, Exceptions

- Lightweight Runtime & Plug-In Mechanism

- Logging Facilities (yes: *SAGA_VERBOSE*)

- SAGA Job Package (Synchronous)

- Proof-of-Concept Local *fork://* Job Adaptor

Compatibility

- Focus on *Pythonic* API (Naming, Properties, ...) but trying to maintain compatibility with the SAGA Python Bindings, e.g.

```python
# separate namespace
import bliss.saga as saga

saga.Url = saga.url
saga.job.Description = saga.job.description

jd = saga.job.description()
jd.set_attribute("Executable", "/usr/bin/bfast") # old-style
jd.arguments = ['xx', 'yy'] # new-style

js = saga.job.Service("fork://localhost")
job = js.create_job(jd)
```

Installation

- Bliss is in the Python Package Index (*PyPI):*
  http://pypi.python.org/pypi/bliss

- Installation via *easy_install* or *pip:*

```
$> easy_install bliss
```

- Or bootstrap in user-space on a remote
  machine including easy_install & virtualenv:

```
$> curl -fsSLk https://raw.github.com/gist/1321016 > \
   pystrap.sh && /bin/sh pystrap.sh -lbliss
```

Contribute

- Yes, Please! It's a Communal Playground...

- Give Feedback / Make Suggestions:
  https://github.com/oweidner/bliss/issues

- Develop: Fork Bliss on GitHub.
  https://github.com/oweidner/bliss

Roadmap

- A Globus GRAM Adaptor

- A PBS via SSH Adaptor

- Tackling Thread-Safety / Metrics Interface ?

- A Minimal File Package ?

Resources

- Preliminary API Documentation: http://oweidner.github.com/bliss/apidoc/

- Wiki & Documentation: https://github.com/oweidner/bliss/wiki

- The *Userspace Bootrapper Script*: https://gist.github.com/1321016

- These Slides: