PyWBEM Python WBEM Client: Overview #2

Karl Schopmeyer
k.schopmeyer@opengroup.org
Andreas Maier
MAIERA@de.ibm.com

April 2016
SNIA SMI plugfest #3

There was an earlier overview At the Feb. 2016 plugfest

Version: 1.0 6 April 2016

Goals for the Presentation

- Present pywbem status and next steps
- Proposal to look beyond the pywbem infrastructure.

PYBEM CLIENT

Status, directions, etc. for the pywbem client product

PyWBEM Client: Overview

- Pure Python code
  - Python 2.6, 2.7, 3.4, 3.5
- Supports DMTF CIM-XML protocol
  - WBEM Client library with a pythonic API
  - Indication listener (experimental)
- Utilities:
  - MOF compiler
  - Command line interface utility
- LGPL 2.1 license

Pywbem Availability

- Client package "pywbem" available in Pypi repository
- Client package available on some Linux distributions
  - Ex. Ubuntu as python-pywbem (v. 0.7.0)
- Directly available from pywbem project on Github:
  - pywbem is a github group with 4 code repositories (pywbem, cimserver, yawn, pyprov) and a doc repository (pywbem.github.io)

    -Download links on PyWBEM github web site:

    `http://pywbem.github.io`
- This presentation concentrates on:
  - **pywbem** – PyWBEM Client project (Python client and related utilities)
  - **pywbem.github.io** – Documentation for the PyWBEM projects

Overall Status Today

- PyWEM Version 0.7.0
  - Considered obsolete
  - Not available on Pypi
  - However, still on many linux distributions
- PyWBEM Client version 0.8
  - Released about 20 March 2016
  - Released 0.8.1, 0.8.2 – Fixes from setup issues
  - Version 0.8.3 in process; setup changes and python 3 ssl
- PyWBEM Client 0.9.0 Next Functional Increment
  - Work started on 0.9.0 – Next significant release

Overall Status Today (cont)

- Version 0.7.0
  - Released 2008 on SourceForge and Pypi by Novell
  - Supports Python 2.6, 2.7
  - Limited tests
  - Distributed on multiple linux platforms
  - Integrated client, compiler, and other experimental components (cimserver, provider, etc. into single repository.
  - See Change log for more detailed information

Version 0.8.0 changes

- Significant code cleanup
- Major documentation update
- Major extensions to test environment
  - Both static and run against servers.
- Support for **Python 3**
- Broke out code into separate repositories
- Add web documentation and separate doc repository
- SSL/Crypto library cleanup
- Install cleanup
- Moved project from SourceForge to Github
- See NEWS file for more details

0.9.0 Work Plan

- Expected Release Date
  - Q2 2016, Hopefully before end of May
- Add SMI required functionality
  - Pull Operations
  - Indication Listener (exists but untested)
- Add many more tests
  - Static and server based tests
- Revamp API documentation
- More cleanup to documentation

0.9.0 Work Plan(cont)

- Correct a number of errors (see github issues)
- Review other code and determine direction
  - Twisted client
- Clean up wbemcli
- Create standalones for wbemcli and mof_compiler

Installation

- Latest release from Pypi (currently 0.8.2)
  - pip install pywbem
  - Available on most Linux OS distributions
- Linux – Install from linux distribution
  - Not current with PyPi releases
- Latest dev. code from Github (0.8.3 dev)
  - –git clone git@github.com:pywbem/pywbem.git
  - –cd pywbem
  - –python setup.py install

Known Issues

- SLP python support
  - No python slp user agent in Pypi
  - Old projects on sourceforge and github
    - They are effectively the same project and represent openslp python interface.
    - Need to find maintainers
    - Are they complete today?
- Other pywbem components
  - Work on providers and a server exist
  - Goal is to determine usability post 0.9.0
- Multiple SSL implementations
  - Python OpenSSL (package ssl) and mtcrypto both exist
  - Today we use one with 2.7 and another with 3
  - Try to sort out a common direction

More Information

- See PyWBEM Client documentation online at:
  - http://pywbem.github.io/pywbem/
- Includes info on:
  - Installation
  - API documentation
  - Usage Tutorial
- Engage with PyWBEM community, for:
  - Reporting issues (github issues)
  - Asking for feature requests (github issues)
  - Contributing (for example from github fork)

EXTENDING BEYOND PYWBEM INFRASTRUCTURE

Goals

- Simplify general WBEM client writing
  - Create a higher level interface for many generic client functions
  - This has been done before in java, powershell
- Simplify SMI client writing
  - Create higher level interface SMI resources
- Provide standard functionality for complex algorithms that can be thoroughly tested

Potential users

- Python application developers
- Provider developers (test tools)
- Admins, etc. – Scripting tools and components for building scripting tools

This is not new ground

- Several groups have created smi client support before
- Powershell cmdlets
  - Microsoft, EMC, Netapp
- Java/Javascript
  - (I think IBM)
- Possibly others
- What makes this different
  - Python
  - OpenSource
  - Open distribution

Overview

- Based on pybem APIs/infrastructure
- Use existing pybem client
- Use existingpybem listener
- Use python based slp (TBD)

Project Proposal

- Open Source.License TBD
- Maintain on github
  - Pywbem github (separate repositories)
- Release as Pypi packages
- Core development team
  - Members of the project with repository write access
- Outsiders contribute through project forks
  - This is github normal approach

Multiple packages

- We see this as multiple packages (importable pieces)
- Tools
  - Useful complete apps as tools
    - Walkers, cli browser, graphical browser, etc.
- WBEM server infrastructure support components
  - Namespaces, profiles, indications, jobs, etc.
- SMI Client support components
  - Profile specific components

Component categories

- Generic client components
  - Server Connection
  - Profile Discovery and central class acquisition
  - Indication management
  - Indication listening, distribution
  - Job Management
  - Slp support
- SMIs specific client target resource elements
  - Storage System
  - Storage Pool
  - Storage Volume
  - Host
  - Etc.

PL7 Not really components but target resource elements.
Peter Lamanna, 3/31/2016

Possible component categories

- Generic client components
  - Server Connection
  - Profile Discovery and central class acquisition
  - Indication management
  - Indication listening, distribution
  - Job Management
  - Slp support on top of openslp python package
- SMIs specific client target resource elements
  - Storage System
  - Storage Pool
  - Storage Volume
  - Host
  - Etc.

PL8 Not really components but target resource elements.
Peter Lamanna, 3/31/2016

Plans

- Plan to start now with infrastructure extensions
  - Experiment with organization and script vs. python internal components architecture.
- Make first prototype available by end of May

We need help

- Tools previously developed and willing to put into open source.
- Developers of previous smi client api/scriptiong projects who would help get the overall architecture right as we move from infrastructure to smi
- Contributors to the work
- There is a lot of work:
  - Pywbem, slp, pywbem tools, pywbem client components
  - We are open to any help we can get.

Questions and Discussion