LCA2006 Systems Administration Miniconf:

System Monitoring with WBEM

Tim Potter
Software Engineer, HP

© 2005 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice
What is WBEM?

• Web
• Based
• Enterprise
• Management
WBEM is:

A protocol for managing network resources.
WBEM is:

Peer and successor to SNMP.
WBEM is:

• XML requests and responses over HTTP
• TCP port 5988 (insecure)
• TCP port 5989 (SSL)
WBEM is:

• An object oriented database
• With remote procedure calls
• With async notifications
  – Indications
  – Like SNMP traps
Scenario

• New widget on your network
• Managed by WBEM
• What do you do?
PyWBEM

• Download and install PyWBEM
• http://pywbem.sourceforge.net
• Start tooling around
Exploring WBEM

• Connect to device

$ python
>>> import pywbem
>>> conn = pywbem.WBEMConnection
    ('widget.foo.com',
     ('username', 'password'))
Exploring WBEM

- Get list of CIM classes

```python
>>> conn.EnumerateClassNames(
    DeepInheritance = True)
['CIM_AFService',
 'CIM_AuthenticationRole,
 ...]
```
Schema Reference

• Latest version available on the DMTF website

http://dmtf.org/standards/cim/cim_schema_v211
Schema Reference

• Class diagrams show CIM class ancestry

• Properties of superclass inherited in subclass
Schema Reference

```
CIM_NumericSensor

CIM_Sensor

CIM_LogicalDevice

CIM_ManagedElement
```
Schema Reference

- MOF files contain property descriptions
- Equivalent to a SNMP MIB
- CIM schema distributed as MOF files
Exploring WBEM

- Some classes that may be present on your device
  - CIM_ComputerSystem
  - CIM_Sensor
  - CIM_PowerSupply
Exploring WBEM

• Get overall status of device

```python
>>> cs = conn.EnumerateInstanceNames('CIM_ComputerSystem')
>>> i = conn.GetInstance(cs[0])
```
Exploring WBEM

• Get overall status of device

```python
>>> print i['OperationalStatus']
[3]

>>> print i['StatusDescriptions']
['Degraded']
```
Exploring WBEM

```python
>>> ps = conn.EnumerateInstances('CIM_PowerSupply')

>>> print(ps[0]['Description'])
'Power Supply 1'
```
Associations

• Can relate managed objects to each other via associations
• e.g. temperature sensor X is connected to CPU Y
• API quite complicated
Associations

• Simple monitoring script

```python
>>> for s in <get sensors>:
    i = conn.GetInstance(s)
>>> print i['OperationalStatus']
```
Indications

• Equivalent of SNMP traps
• Sent as HTTP POST request to a WBEM server
• Subscription API also quite complicated
Example Code

• The PyWBEM home page has lots of examples

• http://pywbem.sf.net/examples/
Key Takeaways

• Mmmm... Takeaway...
• WBEM = CIM Schema + XML + HTTP + SSL
• Object oriented manageability
• Use PyWBEM for exploration
Key Takeaways

• Not present in many devices yet
• Being pushed by big vendors
• Will appear on your radar sometime!