OPENWBEM

An open-source implementation of the CIM and WBEM standards

Enterprise-grade, open-source implementation of WBEM

OpenWBEM is an enterprise-grade, open-source implementation of Web Based Enterprise Management (WBEM), written in C++, suitable for commercial and non-commercial applications. It provides a foundation for development of management frameworks that overcome cross-platform barriers and empower true interoperability. Developers can use OpenWBEM as a management agent and WBEM framework to provide applications for configuration and change management, system health monitoring, and enterprise-wide management functionality.

OpenWBEM Development

OpenWBEM is a complete implementation of WBEM maintained by Vintela, Novell and other volunteers. Dan Nuffer leads the OpenWBEM engineering efforts. While the current version is completely stable and usable, there are many interesting features and optimizations that can still be implemented. As an open-source project, development and participation for OpenWBEM is open to anyone willing to assist and contribute. OpenWBEM is licensed under a BSD licensing model.

Any contribution is welcome, from bug reports, to patches, to new features. If you are interested in joining the OpenWBEM open-source development project, please visit www.openwbem.org.

> “OpenWBEM is great technology that we are pleased to see being adopted from all corners of the industry. With OpenWBEM as the WBEM implementation for VMX—a product at the core of Microsoft's cross-platform management initiatives—and now adopted by Novell for NetWare and SuSE, we anticipate a rapid growth in the acceptance, proliferation, and success of the standard. It truly extends CIM to all platforms.”
>
> *Matt Peterson, Chief Technology Officer, Vintela*

Related OpenWBEM Projects

- **Owperlprovider**
  http://jason.long.name/owperl/

  “A new Perl provider interface for OpenWBEM. This lets you write OpenWBEM providers in Perl.”

- **Kim-browser**
  http://sourceforge.net/projects/kde-cim

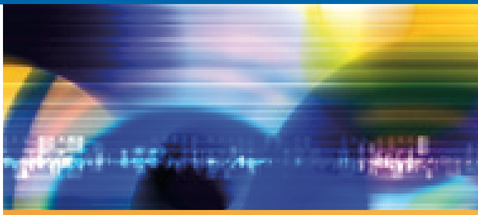
  “The KDE CIM Browser is a KDE based frontend to management brokers using the CIM-XML protocol. The tool will be useful to view and modify CIM class hierarchies as well as instance and association structures.” The KDE CIM Browser uses the OpenWBEM client API.

- **CimNavigator**
  http://cimnavigator.com/

  “A Java client application that can be used to graphically browse CIM objects and their associations in various CIMOMs.”

- **SBLIM**
  http://www-124.ibm.com/sblim/

  A lot of the SBLIM Linux providers work with OpenWBEM NPI & CMPI provider interfaces. “SBLIM (pronounced “sublime”), the Standards Based Linux Instrumentation for Manageability is an IBM Open Source project, intended to enhance the manageability of GNU/Linux systems.”



OpenWBEM enables application developers to architect management frameworks that provide cross-platform interoperability. Some of the main features and capabilities of OpenWBEM include:

- **CIM Server** – The DMTF's *CIM Operations over HTTP 1.1* is fully supported. CIM 2.2 is supported, including embedded instances, which are necessary for proper indication support.
- **CIM Providers Support** – Writing providers is simple. Base class adapters are provided for the most common situations which reduces development time and provider size. Helper functions are available that make it easy to correctly write providers. Other CIM Provider support includes:
  - Providers run in a transactional environment, making it much easier to write a robust provider free from deadlocks and race conditions.
  - OpenWBEM C++ Providers are automatically detected, so there is no need to use the provider qualifier or registration schema.
  - Instance
  - Secondary Instance
  - Method
  - Associator
  - Indication
  - Indication Export
- **Supported Provider Interface Types:**
  - C++
  - CMPI. Thanks to Jon Carey/Novell
  - NPI Provider Interface
  - Perl based on NPI
  - Perl using SWIG, created by Jason Long. Located at (http://jason.long.name/owperl/)
- **Indications and CIM Listener support.** Both lifecycle and alert indications are supported.
- **WQL Level 2 support for server and client**
- **Access Control Lists**
- **Discovery via SLP Integration on the CIMOM and client side using any RFC 2614 compliant SLP implementation (e.g. OpenSLP)**
- **Support for a binary protocol that is much more efficient than CIM/XML**
- **http deflate compression with zlib. (Using the binary protocol and compression is very bandwidth efficient.)**
- **http chunking & trailers allows OpenWBEM to transfer large transactions (e.g. enumClasses on the entire schema) with no buffering.** Memory usage stays constant. Even if the client does not support chunking and trailers, large requests are buffered on disk to save memory.
- **The owcimomd daemon has been designed to be non-intrusive and have a small footprint.** Many features are loaded as shared libraries so you can pick and choose what features you really need and not waste memory with unused code. Optional features include:
  - Indication support
  - WQL library
  - Authentication modules
  - Authorization modules
  - Provider interfaces
  - Providers are unloaded from memory if they are not used for a configurable amount of time
  - SLP support is implemented as a provider
  - Request handlers (CIM/XML or binary) can be loaded/unloaded on demand with a configurable unload timeout
  - HTTP compression
  - HTTP digest authorization
  - HTTPS (SSL)
  - Functional Profiles (Association Traversal, Instance Manipulation, etc.)

Available Providers:

- CIM_Namespace - Allows manipulation of namespaces.
- OpenWBEM_ObjectManager : CIM_ObjectManager - Represents owcimomd.
- CIM_NamespaceInManager - Association between CIM_Namespace and CIM_ObjectManager
- SLP advertisement
- __Namespace - Allows manipulation of namespaces for legacy CIM Clients using the deprecated __Namespace class.
- IBM has a lot of system level providers available via the SBLIM project. All these NPI & CMPI providers work with OpenWBEM
- Example providers show how to implement providers. These providers instrument processes, RPMs, and laptop batteries. These make for good demos as well.

Additional Software:

- C++ CIM model & API
- WBEM Client API
- MOF Compiler
- MOF API & library can be utilized from any application. Includes an API to convert mof text into a C++ CIM class/instance without talking to a cimom.
- WQL command-line utility
- WQL client library. Use WQL with servers which don't support it or don't support your particular queries.

For more information
Visit the OpenWBEM Web site
www.openwbem.org