



**WBEM**

**SOLUTIONS**

*Bringing IT Together!*



# **WBEM Solutions**

## **SDK Pro**

### **3.0**



# Overview

APIs

Examples

Documentation

Utilities

CIM

WBEM

Code Generation

J WBEM Server



# APIs



# APIs

JSR48

SLP

CMPI



## JSR48

Client APIs for all WBEM operations

Provider APIs

Instance, Association, Indication, Method

Listener APIs

Supports CIM-XML and WS-Management

Supports Pull Operations

Full API Documentation included in SDK



# JSR48 Client Examples

Operations

Initiative Use Cases

Code Generation

Profiles



# JSR48 Client Examples Operations

Fully operational example client code for the following types of WBEM operations

Class

Indication

Instance

Extrinsic Methods

QualifierType

Query



# JSR48 Client Examples Initiative Use Cases

Implementations of various Use Cases from selected profiles of the following Management Initiatives

CDM

DASH

SMI



OpenSLP Implementation

[www.openslp.org](http://www.openslp.org)

C API.

Defines APIs for client discovery of services advertised by SLP

Defines APIs for service registration

Example included for client discovery



Standard from The Open Group  
(<http://www.opengroup.org>)

Used to create C-based WBEM providers

Defines an API for each CIM data type

Defines a Provider API

SDK Pro includes CMPI documentation



# Utilities



## CIM Utilities

mofcomp

mofstyle

mofinclude

cimmorph

mofmorph

schemadiff

schematest



## **mofcomp** **Description**

Command Line utility that parses MOF files, checks syntax, and compiles CIM classes, qualifiers and instances into a CIM repository

Can run in standalone mode

Can run remotely against a target WBEM Server using CIM-XML

# mofcomp CLI Syntax

```
mofcomp [options] MOFFILE
```

## Usage Options:

|                 |  |
|-----------------|--|
| -c              | Syntax Checking Only                         |
| -d              | Display debug information                    |
| -h              | Display this usage statement                 |
| -i <name>       | Interop namespace                            |
| -kf <file>      | KeyStore File (Mutual Authentication)        |
| -kp <password>  | KeyStore Password (Mutual Authentication)    |
| -l <locales>    | List of locales in priority order            |
| -n <name>       | Namespace                                    |
| -o <dir>        | Directory to output files                    |
| -p <credential> | WBEM Server credential (e.g. password)       |
| -q              | Quiet mode                                   |
| -s <server>     | WBEM Server URL                              |
| -tf <file>      | TrustStore File (Mutual Authentication)      |
| -u <principal>  | WBEM Server principal (e.g. username)        |
| -v              | Display the version information              |
| -wpp            | WBEM Protocol (e.g. CIM-XML, ...)            |
| -m              | Allow modification of elements if they exist |



# mofcomp

## Example: Standalone

```
> mofcomp -o tmp -n cimv220 cim_schema_2.20.1.mof
```

```
Created namespace: https://localhost/interop
Creating qualifier type Association
Creating qualifier type Indication
...
Creating qualifier type UnsupportedValues
Creating class CIM_ManagedElement
Creating class CIM_ManagedSystemElement
Creating class CIM_MethodParameters
...
Creating class CIM_ComputerSystemMappedIO
Creating class CIM_CASignsPublicKeyCertificate
Creating class CIM_CASignsPublicKeyCertificate
Compilation succeeded.
71 Qualifier Types.
1277 Classes.
```

```
> dir /b tmp
Logfile.12
Snapshot.12
store
Version_Number
```

# mofcomp

## Example: Remote

```
> mofcomp -s https://192.168.10.20 -n cimv220 cim_schema_2.20.1.mof
```

```
Created namespace: https://localhost/interop  
Creating qualifier type Association  
Creating qualifier type Indication  
...  
Creating qualifier type UnsupportedValues  
Creating class CIM_ManagedElement  
Creating class CIM_ManagedSystemElement  
Creating class CIM_MethodParameters  
...  
Creating class CIM_ComputerSystemMappedIO  
Creating class CIM_CASignsPublicKeyCertificate  
Creating class CIM_CASignsPublicKeyCertificate  
Compilation succeeded.  
71 Qualifier Types.  
1277 Classes.
```

```
>
```



# mofstyle Description

Command Line utility that processes CIM classes defined in a MOF file and reformats the text according to DMTF Style guidelines

Processes all files with .mof suffix in a target directory and all subdirectories



# mofstyle CLI Syntax

```
mofstyle [options] directory
```

## Usage Options:

- h Display this usage statement
- q Quiet mode
- v Display the version information



# mofstyle Example:

```
> mofstyle ..\..\standards\dmf\schema\2.20.1\final\mof
```

```
Processing ..\..\standards\dmf\schema\2.20.1\final\mof\Application\CIM_Action.mof
```

```
...
```

```
Processing ..\..\standards\dmf\schema\2.20.1\final\mof\Core\CIM_BasedOn.mof
```

```
...
```

```
Processing ..\..\standards\dmf\schema\2.20.1\final\mof\Device\CIM_AlarmDevice.mof
```

```
...
```

```
Processing ..\..\standards\dmf\schema\2.20.1\final\mof\Network\CIM_Network.mof
```

```
...
```

```
Processing ..\..\standards\dmf\schema\2.20.1\final\mof\User\CIM_UsersAccount.mof
```

```
...
```

```
MOFStyle completed successfully.
```



## **mofinclude Description**

Command Line utility that generates a MOF file containing #pragma include statements to other MOF files in a schema.



# mofinclude CLI Syntax

```
mofinclude [options] directory outfile
```

## Usage Options:

- h Display this usage statement
- v Display the version information
- cf Copyright File
- l locale pragma
- qf Qualifier Include Files

# mofinclude

## Example:

```
> mofinclude -cf copyright.txt -l en_US -qf  
  qualifiers.mof,qualifiers_optional.mof  
  ..\..\standards\dmtof\schemata\2.15\final\mof cimv215.mof
```

Reading

```
  ..\..\standards\dmtof\schemata\2.15\final\mof\Application\CIM_Actio  
  n.mof
```

...

Reading

```
  ..\..\standards\dmtof\schemata\2.15\final\mof\User\CIM_CASignsPubl  
  icKeyCertificate.mof
```

```
> type cimv215.mof
```

```
Copyright 2007 WBEM Solutions, Inc. All rights reserved.
```

```
#pragma locale ("en_US")
```

```
#pragma include ("qualifiers.mof")
```

```
#pragma include ("qualifiers_optional.mof")
```

```
#pragma include ("Core/CIM_ManagedElement
```

...

```
#pragma include("User/CIM_CASignsPublicKeyCertificate.mof")
```



# **schemadiff**

## **Description**

Command Line utility that compares two schemas and generates a report describing the differences between the two.

This utility is useful in identifying the changes made to the previous version of MOF files.



# schemadiff CLI Syntax

```
schemadiff [options] schema1 schema2
```

## Usage Options:

- h Display this usage statement
- o <dir> Directory to output files
- q Quiet mode
- v Display the version information

# schemadiff

## Example: command

```
> schemadiff -o diff ..\..\standards\dmf\schema\2.14\mof\cimv214.mof  
    ..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
```

**Processing Association**

**Processing Indication**

:

**Processing CIM\_CAHasPublicCertificate**

**Processing CIM\_CASignsPublicKeyCertificate**

**SchemaDiff generated 'schemadiff.html'.**

```
>cd diff
```

```
>dir /b *.html
```

**all\_classes.html**

**CIM\_AbstractIndicationSubscription.html**

:

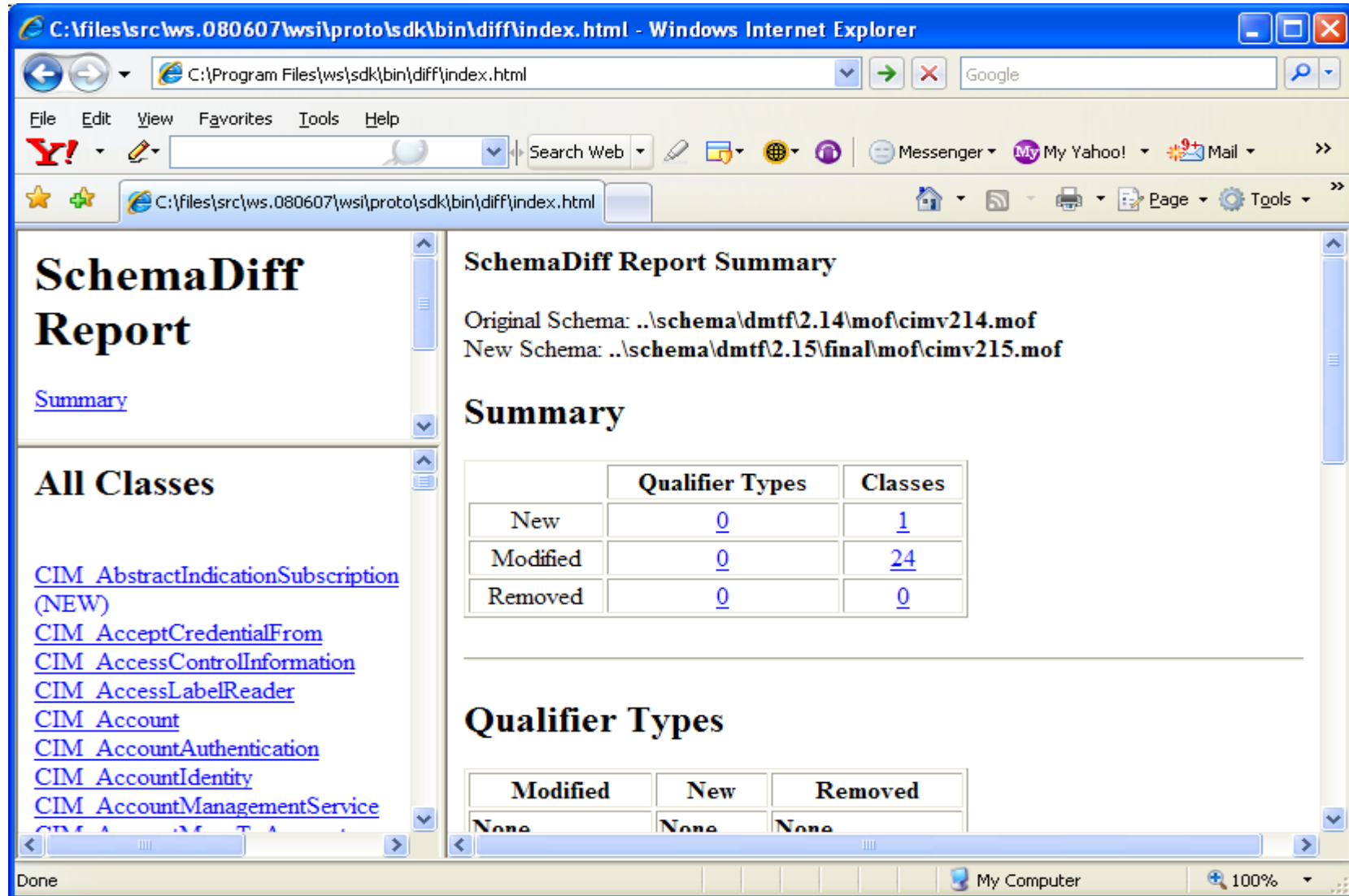
**index.html**

:

**PRS\_Transaction.html**

# schemadiff

## Example: report



The screenshot shows a web browser window displaying a SchemaDiff Report. The report compares two schemas: the original schema (CIMV214.mof) and the new schema (CIMV215.mof). The report is divided into sections: Summary, All Classes, and Qualifier Types.

**SchemaDiff Report Summary**

Original Schema: `..\schema\dmf\2.14\mof\cimv214.mof`  
 New Schema: `..\schema\dmf\2.15\final\mof\cimv215.mof`

**Summary**

|          | Qualifier Types | Classes   |
|----------|-----------------|-----------|
| New      | <u>0</u>        | <u>1</u>  |
| Modified | <u>0</u>        | <u>24</u> |
| Removed  | <u>0</u>        | <u>0</u>  |

**All Classes**

- [CIM AbstractIndicationSubscription \(NEW\)](#)
- [CIM AcceptCredentialFrom](#)
- [CIM AccessControlInformation](#)
- [CIM AccessLabelReader](#)
- [CIM Account](#)
- [CIM AccountAuthentication](#)
- [CIM AccountIdentity](#)
- [CIM AccountManagementService](#)

**Qualifier Types**

| Modified | New  | Removed |
|----------|------|---------|
| None     | None | None    |



## **schematest Description**

Command Line utility that compares a new schema to an older schema and generates a report identifying any incompatible changes that may have been made.



# schematest CLI Syntax

```
schematest [options] schema1 schema2
```

## Usage Options:

- h Display this usage statement
- v Display the version information

# schematest

## Example: command

```
> schematest ..\..\standards\dmf\schema\2.14\mof\cimv214.mof  
    ..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
```

Loading ..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof ...

Loading ..\..\standards\dmf\schema\2.14\mof\cimv214.mof ...

**ERROR[CIM\_IndicationSubscription]: The Experimental qualifier can not be added in a minor release.**

**WARNING[CIM\_IndicationSubscription.OnFatalErrorPolicy]: The property CIM\_IndicationSubscription.OnFatalErrorPolicy was removed from the class, but is inherited**

**- it may have been moved up to a super class.**

**WARNING[CIM\_IndicationSubscription.OtherOnFatalErrorPolicy]:**

**The property**

**CIM\_IndicationSubscription.OtherOnFatalErrorPolicy was removed from the class, but is inherited - it may have been moved up to a super class.**

...



# cimmorph Description

Command Line utility that processes CIM class defined in the repository of a target WBEM server and converts them to the following formats:

**HTML**

**XML based on the xmlCIM mapping**

**XSD based on the WS-CIM mapping**

**Provider source code template written in Java using the JSR48 WBEM API**

**Client source code template written in Java using the JSR48 WBEM API**

**Provider source code template written in C using the CMPI API**



# cimmorph CLI Syntax

`cimmorph [options] Format Mapping classname`

## Usage Options:

|                    |  |
|--------------------|--|
| -h                 | Display this usage statement                     |
| -kf <file>         | KeyStore File (Mutual Authentication)            |
| -kp <password>     | KeyStore Password (Mutual Authentication)        |
| -o <dir>           | Directory to output files                        |
| -p <credential>    | WBEM Server credential (e.g. Password)           |
| -ph <host>         | HTTP Proxy Host                                  |
| -pp <port>         | HTTP Proxy Port                                  |
| -ru <principal>    | WBEM Server Role principal (e.g. Role to assume) |
| -rp <credential>   | WBEM Server Role credential (e.g. Password)      |
| -q                 | Quiet Mode                                       |
| -s <server>        | WBEM Server URL                                  |
| -tf <file>         | TrustStore File (Mutual Authentication)          |
| -u <principal>     | WBEM Server principal (e.g. username)            |
| -v                 | Display the version information                  |
| -wp                | WBEM Protocol (e.g. CIM-XML, WS-Management, ...) |
| -cp <name=value,*> | Custom Properties                                |

## Commands:

|           |   |
|-----------|---|
| Format    | Format of the output (XSD, XML, MOF, ...) |
| Mapping   | Mapping to use (WS-CIM, xmlCIM, ...)      |
| classname | The class name to be mapped               |



# cimmorph

## Example: Generate HTML

```
> cimmorph -s https://localhost/interop -u root -p  
    pokey -o interop\html wsi HTML cim_managedelement
```

**Generating HTML...**

**Removing Temporary Files...**

**CIMMorph completed successfully.**

```
> cd interop\html
```

```
> dir /b
```

**allclasses.html**

**CIM\_ManagedElement.html**

**classdetail.html**

**index.html**

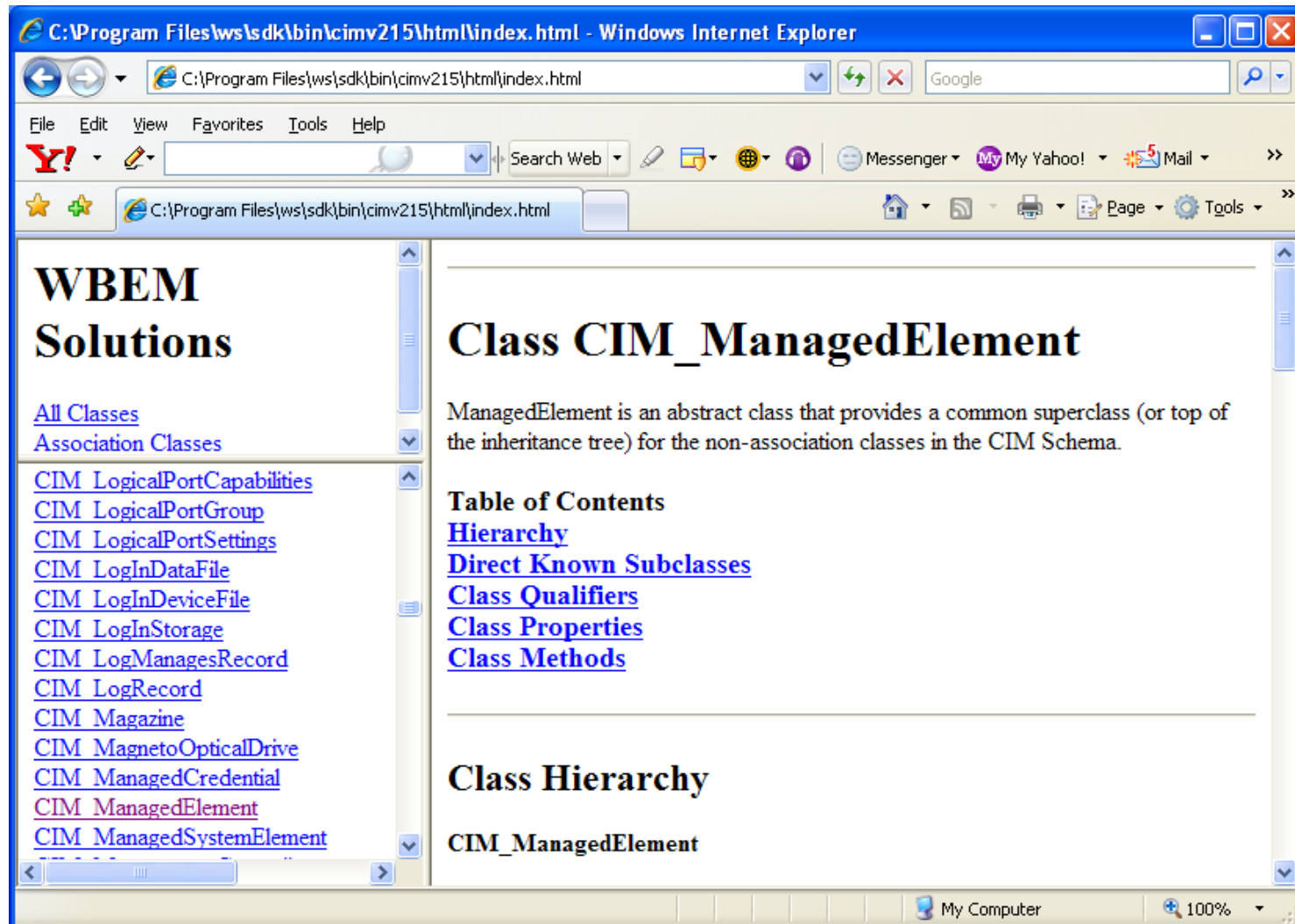
**mof.html**

**schema.html**

```
>
```

# cimmorph

## Example: Generate HTML



C:\Program Files\ws\sd\bin\cimv215\html\index.html - Windows Internet Explorer

C:\Program Files\ws\sd\bin\cimv215\html\index.html

File Edit View Favorites Tools Help

Y! Search Web Messenger My My Yahoo! Mail

C:\Program Files\ws\sd\bin\cimv215\html\index.html

## WBEM Solutions

[All Classes](#)  
[Association Classes](#)

[CIM\\_LogicalPortCapabilities](#)  
[CIM\\_LogicalPortGroup](#)  
[CIM\\_LogicalPortSettings](#)  
[CIM\\_LogInDataFile](#)  
[CIM\\_LogInDeviceFile](#)  
[CIM\\_LogInStorage](#)  
[CIM\\_LogManagesRecord](#)  
[CIM\\_LogRecord](#)  
[CIM\\_Magazine](#)  
[CIM\\_MagnetoOpticalDrive](#)  
[CIM\\_ManagedCredential](#)  
[CIM\\_ManagedElement](#)  
[CIM\\_ManagedSystemElement](#)

## Class CIM\_ManagedElement

ManagedElement is an abstract class that provides a common superclass (or top of the inheritance tree) for the non-association classes in the CIM Schema.

### Table of Contents

[Hierarchy](#)  
[Direct Known Subclasses](#)  
[Class Qualifiers](#)  
[Class Properties](#)  
[Class Methods](#)

### Class Hierarchy

CIM\_ManagedElement

My Computer 100%

# cimmorph

## Example: Generate XML

```
> cimmorph -s https://localhost/interop -u root -p  
pokey -o interop\xml xml xmlCIM cim_managedelement
```

**CIMMorph completed successfully.**

```
> cd interop\xml
```

```
> dir /b
```

**all\_classes.xml**

**CIM\_ManagedElement.xml**

**class\_index.xml**

```
> type CIM_ManagedElement.xml
```

```
<?xml version="1.0"?>  
<CIM CIMVERSION="2.3" DTDVERSION="2.2">  
  <DECLARATION>  
    <DECLGROUP>  
      <VALUE.OBJECT>  
        <CLASS NAME="CIM_ManagedElement">  
..  
      </CLASS>  
    </VALUE.OBJECT>  
  </DECLGROUP>  
</DECLARATION>  
</CIM>
```



# cimmorph

## Example: Generate XSD

```
> cimmorph -s https://localhost/interop -u root -p  
    pokey -o interop\xsd xsd WS-CIM cim_managedelement
```

**CIMMorph completed successfully.**

```
> cd interop\xsd
```

```
> dir /b
```

**CIM\_ManagedElement.xsd**

```
> type CIM_ManagedElement.xsd
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--Generated by WBEM Solutions, Inc. SDKPro 2.0.0 (Beta)-->
```

```
...
```

```
<xs:complexType name="CIM_ManagedElement_Type">
```

```
  <xs:sequence>
```

```
    <xs:element minOccurs="0" ref="class:Caption"/>
```

```
    <xs:element minOccurs="0" ref="class:Description"/>
```

```
    <xs:element minOccurs="0" ref="class:ElementName"/>
```

```
    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other"  
      processContents="lax"/>
```

```
  </xs:sequence>
```

```
  <xs:anyAttribute namespace="##any" processContents="lax"/>
```

```
</xs:complexType>
```

```
</xs:schema>
```



# cimmorph

## Example: Generate Java Client

```
> cimmorph -s https://localhost/interop -u root -p  
    pokey -o interop\jsr48\client client Java  
    CIM_ObjectManager
```

**CIMMorph completed successfully.**

```
> cd interop\jsr48\client
```

```
> dir /b
```

**CIM\_Bean.java**

**CIM\_BeanImpl.java**

**CIM\_ObjectManagerBean.java**

**CIM\_ObjectManagerBeanImpl.java**



# cimmorph

## Example: Generate Java Provider

```
> cimmorph -s https://localhost/interop -u root -p  
    pokey -o interop\jsr48\provider provider Java  
    CIM_ObjectManager
```

**CIMMorph completed successfully.**

```
> cd interop\jsr48\provider
```

```
> dir /b
```

**CIM\_ObjectManagerProvider.java**

**CloseableIteratorImpl.java**



# cimmorph

## Example: Generate C Provider

```
> cimmorph -s https://localhost/interop -u root -p  
    pokey -o interop\cmpi\provider cmpi C  
    CIM_ObjectManager
```

**CIMMorph completed successfully.**

```
> cd interop\cmpi\provider
```

```
> dir /b
```

```
CIM_ObjectManager.c
```

```
CIM_ObjectManager.h
```



# mofmorph Description

Command Line utility that processes CIM classes defined in MOF files and converts them to the following formats:

**HTML**

**XML based on the xmlCIM mapping**

**XSD based on the WS-CIM mapping**

**Provider source code template written in Java using the JSR48 WBEM API**

**Client source code template written in Java using the JSR48 WBEM API**

**Provider source code template written in C using the CMPI API**



# mofmorph CLI Syntax

`mofmorph [options] Format Mapping moffile`

## Usage Options:

|                                       |                                   |
|---------------------------------------|-----------------------------------|
| <code>-h</code>                       | Display this usage statement      |
| <code>-o &lt;dir&gt;</code>           | Directory to output files         |
| <code>-q</code>                       | Quiet mode                        |
| <code>-v</code>                       | Display the version information   |
| <code>-cp &lt;name=value,*&gt;</code> | Custom Properties                 |
| <code>-r &lt;dir&gt;</code>           | Use Existing Repository Directory |

## Commands:

|           |   |
|-----------|---|
| Format    | Format of the output (XSD, XML, MOF, ...) |
| Mapping   | Mapping to use (WS-CIM, xmlCIM, ...)      |
| classname | The class name to be mapped               |



# mofmorph

## Example: Generate HTML

```
> mofmorph -o cimv215\html wsi HTML  
   ..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
```

Processing CIM\_DiagnosticResult

...

Processing CIM\_CASignsPublicKeyCertificate

Generating HTML...

Removing Temporary Files...

MOFMorph completed successfully.

```
> cd cimv215\html
```

```
> dir /b
```

allclasses.html

CIM\_AcceptCredentialFrom.html

... :

index.html

...

PRS\_Transaction.html

schema.html



# mofmorph

## Example: Generate XML

```
> mofmorph -o cimv215xml xml xmlCIM  
..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
```

**Processing CIM\_DiagnosticResult**

:

**Processing CIM\_CASignsPublicKeyCertificate**  
**MOFMorph completed successfully.**

```
> cd cimv215\xml
```

```
> dir /b
```

**all\_classes.xml**

**CIM\_AcceptCredentialFrom.xml**

**CIM\_AccessControlInformation.xml**

**CIM\_AccessLabelReader.xml**

:

**PRS\_Statement.xml**

**PRS\_StatementFeature.xml**

**PRS\_Transaction.xml**



# mofmorph

## Example: Generate XSD

```
> cimmorph -s https://localhost/interop -u root -p pokey  
-o interop\xsd xsd WS-CIM cim_managedelement
```

**CIMMorph completed successfully.**

```
> cd interop\xsd
```

```
> dir /b
```

**all\_classes.xsd**

**CIM\_AcceptCredentialFrom.xsd**

**CIM\_AccessControllInformation.xsd**

**CIM\_AccessLabelReader.xsd**

**:**

**PRS\_Statement.xsd**

**PRS\_StatementFeature.xsd**

**PRS\_Transaction.xsd**



# mofmorph

## Example: Generate Java Client

```
> mofmorph -o cimv215\jsr48\client client Java
..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
Processing CIM_DiagnosticResult
:
Processing CIM_CASignsPublicKeyCertificate
MOFMorph completed successfully.
> cd cimv215\jsr48\client
> dir /b *.java
CIM_Bean.java
CIM_BeanImpl.java
CIM_AcceptCredentialFromBean.java
CIM_AcceptCredentialFromBeanImpl.java
:
PRS_TransactionBean.java
PRS_TransactionBeanImpl.java
```

# mofmorph

## Example: Generate Java Provider

```
> mofmorph -o cimv215\jsr48\provider provider Java
..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
Processing CIM_DiagnosticResult
:
Processing CIM_CASignsPublicKeyCertificate
MOFMorph completed successfully.
> cd cimv215\jsr48\provider
> dir /b *.java
CIM_AcceptCredentialFromProvider.java
CIM_AccessControlInformationProvider.java
:
PRS_StatementFeatureProvider.java
PRS_TransactionProvider.java
```



# mofmorph

## Example: Generate C Provider

```
> mofmorph -o cimv215\cmpi\provider cmpi C
..\..\standards\dmf\schema\2.15\final\mof\cimv215.mof
Processing CIM_DiagnosticResult
:
Processing CIM_CASignsPublicKeyCertificate
MOFMorph completed successfully.
> cd cimv215\cmpi\provider
> dir /b *.c
CIM_AcceptCredentialFrom.c
CIM_AccessControlInformation.c
:
PRS_Statement.c
PRS_StatementFeature.c
```



# WBEM Utilities

wbemop

wbemmethod

wbemind

wbemextract

cimworkshop



## **wbemop** **Description**

Command Line utility that allows a user to perform WBEM operations on a target WBEM server

Metadata (e.g. GetClass)

Data (e.g. EnumerateInstances, ExecQuery)

Association Traversal (e.g. References)

Pull Operations

Supports CIM-XML and WS-Management



# wbemop CLI Syntax

wbemop [options] CIMOBJECTPATH COMMAND [attrs]

## Usage Options:

|                       |  |
|-----------------------|--|
| -ac                   | Association Class argument                                 |
| -ar                   | Association Role argument                                  |
| -co                   | Include Class Origin                                       |
| -d                    | Display XML debug information                              |
| -dh                   | Deep Inheritance off                                       |
| -f <filter>           | Filter for query execution                                 |
| -h                    | Prints this usage statement                                |
| -kf <file>            | KeyStore File (Mutual Authentication)                      |
| -kp <password>        | KeyStore Password (Mutual Authentication)                  |
| -l <locales>          | The comma separated list of locales in preference order    |
| -lo                   | Include local properties only                              |
| -nq                   | Do not include qualifiers                                  |
| -p <credential>       | CIM Server credential(e.g. Password)                       |
| -ph <host>            | HTTP Proxy Host  |
| -pl <prop>,<prop>,... | Comma-separated property list                              |
| -ql <query language>  | Query Language to use for executing queries                |
| -rc                   | Result class argument                                      |
| -rp <principal>       | Role Principal (e.g. role name)                            |
| -ru <credential>      | Role Credential (e.g. password)                            |
| -rr                   | Result Role argument                                       |
| -s <server>           | WBEM Server URL  |
| -tf <file>            | TrustStore File (Mutual Authentication)                    |
| -u <principal>        | WBEM Server principal(e.g. username)                       |
| -v                    | Prints the version   |
| -wp                   | WBEM Protocol (e.g.CIM-XML,WS-Management,...)              |
| -x                    | Prints the results in XML                                  |
| -dm                   | Display class methods for class when using a property list |



# wbemop CLI Syntax (cont.)

## Command:

```
ci      Create an Instance
dc      Delete a Class
di      Delete an Instance
dq      Delete a Qualifier Type Declaration
ec      Enumerate Classes
ei      Enumerate Instances
eq      Enumerate Qualifier Types Declarations
gc      Get a Class
gi      Get an Instance
gq      Get a Qualifier Type Declaration
mi      Modify an Instance
nc      Enumerate Class Names
ni      Enumerate Instance Names
as      Associators
an      AssociatorNames
rs      References
rn      ReferenceNames
qe      Execute the supplied/default query.
pan     Pull Associator Names
pas     Pull Associators
pei     Pull Enumerate Instance
pni     Pull Enumerate Instance Names
prn     Pull Reference Names
prs     Pull References
```

## Attrs:

```
prop=value[,prop=value]
```



# wbemop CLI Syntax (cont.)

## Command:

```
ci      Create an Instance
dc      Delete a Class
di      Delete an Instance
dq      Delete a Qualifier Type Declaration
ec      Enumerate Classes
ei      Enumerate Instances
eq      Enumerate Qualifier Types Declarations
gc      Get a Class
gi      Get an Instance
gq      Get a Qualifier Type Declaration
mi      Modify an Instance
nc      Enumerate Class Names
ni      Enumerate Instance Names
as      Associators
an      AssociatorNames
rs      References
rn      ReferenceNames
qe      Execute the supplied/default query.
pan     Pull Associator Names
pas     Pull Associators
pei     Pull Enumerate Instance
pni     Pull Enumerate Instance Names
prn     Pull Reference Names
prs     Pull References
```

## Attrs:

```
prop=value[,prop=value]
```



# wbemop

## Example: GetInstance

```
> wbemop -s https://localhost/interop -u user1 -p pass1
  WBEMSolutions_CIMXMLObjectManagerAdapter.Name="CIM-
  XML",SystemCreationClassName="WBEMSolutions_ComputerSy
  stem",SystemName="192.168.2.3",CreationClassName="WBEMS
  olutions_CIMXMLObjectManagerAdapter" gi
```

```
instance of WBEMSolutions_CIMXMLObjectManagerAdapter {
  classPath = null;
  PortNumber = 5988;
  ProtocolType = "http";
  Name = "CIM-XML";
  ElementName = "CIM-XML";
  Handle = "org.wbemservices.wbem.cimom.adapters.client.http.HttpService";
  AdapterType = 2;
  SystemCreationClassName = "WBEMSolutions_ComputerSystem";
  SystemName = "192.168.5.20";
  CreationClassName = "WBEMSolutions_CIMXMLObjectManagerAdapter";
  ...
  OperationalStatus = {2};
  Caption = "CIM-XML Client Adapter";
  Description = "WBEM Solutions CIM-XML Adapter.";
};
>
```

# wbemop

## Example: EnumerateInstanceNames

```
> wbemop -s https://localhost/interop -u user1 -p pass1  
CIM_ObjectManagerAdapter ni
```

```
//localhost/interop:WBEM Solutions_ObjectManagerAdapter.Name="jsr48",SystemCreationClassName="WBEM Solutions_ComputerSystem",SystemName="192.168.5.20",CreationClassName="WBEM Solutions_ObjectManagerAdapter"
```

```
//localhost/interop:WBEM Solutions_ObjectManagerAdapter.Name="IH:CIM-XML",SystemCreationClassName="WBEM Solutions_ComputerSystem",SystemName="192.168.5.20",CreationClassName="WBEM Solutions_ObjectManagerAdapter"
```

```
//localhost/interop:WBEM Solutions_ObjectManagerAdapter.Name="java",SystemCreationClassName="WBEM Solutions_ComputerSystem",SystemName="192.168.5.20",CreationClassName="WBEM Solutions_ObjectManagerAdapter"
```

```
//localhost/interop:WBEM Solutions_CIMXMLSObjectManagerAdapter.Name="CIM-XML-SSL",SystemCreationClassName="WBEM Solutions_ComputerSystem",SystemName="192.168.5.20",CreationClassName="WBEM Solutions_CIMXMLSObjectManagerAdapter"
```



## **wbemmethod Description**

Command Line utility that allows a user to perform extrinsic methods on a target WBEM server  
Supports CIM-XML and WS-Management



# wbemmethod CLI Syntax

`WBEMMethod [options] objectpath method`

## Usage Options:

|  |  |
|--|--|
| <code>-ai [name=value,*]</code>                | Input arguments                                  |
| <code>-ao &lt;prop&gt;,&lt;prop&gt;,...</code> | Output arguments                                 |
| <code>-d</code>                                | Display debug information                        |
| <code>-h</code>                                | Prints this usage statement                      |
| <code>-kf &lt;file&gt;</code>                  | KeyStore File (Mutual Authentication)            |
| <code>-kp &lt;password&gt;</code>              | KeyStore Password (Mutual Authentication)        |
| <code>-p &lt;credential&gt;</code>             | WBEM Server credential (e.g.password)            |
| <code>-ph &lt;host&gt;</code>                  | HTTP Proxy Host                                  |
| <code>-pp &lt;port&gt;</code>                  | HTTP Proxy Port                                  |
| <code>-rp &lt;credential&gt;</code>            | WBEM Server Role credential (e.g. password)      |
| <code>-ru &lt;principal&gt;</code>             | WBEM Server Role principal (e.g. role to assume) |
| <code>-s &lt;server&gt;</code>                 | WBEM Server URL                                  |
| <code>-tf &lt;file&gt;</code>                  | TrustStore File (Mutual Authentication)          |
| <code>-u &lt;principal&gt;</code>              | WBEM Server principal (e.g.username)             |
| <code>-v</code>                                | Display the version information                  |
| <code>-w</code>                                | WBEM Protocol (e.g. CIM-XML, WS-Management)      |



# wbemmethod

## Example:

```
> wbemmethod -s https://localhost/interop -u Administrator -p admin  
WBEMSolutions_CIMXMLObjectManagerAdapter.Name="CIM-  
XML",SystemCreationClassName="WBEMSolutions_ComputerSystem",S  
ystemName="192.168.1.2",CreationClassName="WBEMSolutions_CIMXM  
LObjectManagerAdapter" StartService
```

Return Value: 0

>



# wbemmethod Example:

```
> wbemind -s https://localhost/interop -u root -p pokey -c Test_Test -ci
```

Listening for indications:

Hit "Enter" key at any time to quit.

\*\*\*\*\*

Start : Indication of Instance Creation

\*\*\*\*\*

```
instance of cim_instcreation {  
    [EmbeddedObject(true)  
    ]  
    SourceInstance = instance of Test_Test {  
        stringKey = "Brady";  
        intKey = 12;  
        stringProp = "Patriots";  
        boolProp = null;  
        arrayProp = {14, 2};  
};
```

```
};
```

\*\*\*\*\*

End : Indication of Instance Creation

\*\*\*\*\*

Hit "Enter" key at any time to quit.

Removing event subscriptions...

```
>
```



## wbemind Description

Command Line utility that allows a user to specify an indication filter

Listens for indications until stopped by the user

Supports life cycle and process indications

User can create a listener for

Class Indications

Instance Indications

Method Indications

Currently only supports CIM-XML

Future revision will support WS-Management



# wbemind CLI Syntax

wbemind [-options]

## Usage Options:

- d Display debug information
- f <filter> Free form indication filter.
- h Print out this message.
- i <name> Interop namespace
- kf <file> KeyStore File (Mutual Authentication)
- kp <password> KeyStore Password (Mutual Authentication)
- l locales List of locales in priority order.
- n <name> Namespace in which to listen for indications
- p <credential> WBEM Server credential (e.g. Password)
- ph <host> HTTP Proxy Host
- pp <port> HTTP Proxy Port
- ru <principal> WBEM Server Role principal(e.g.role to assume)
- rp <credential> WBEM Server Role credential (e.g. password)
- s <server> WBEM Server URL
- u <principal> WBEM Server Principal (e.g. username).
- v Display the version information
- wp WBEM Protocol (e.g. CIM-XML, WS-Management, ...)

# wbemind CLI Syntax (cont.)

|                 |  |
|-----------------|--|
| -x              | Display the results in XML                           |
| -c              | Classname for instance indications                   |
| -ps             | Create permanent indication subscription             |
| -cc             | Listen for class creation indications.               |
| -ci             | Listen for instance creation indications.            |
| -dc             | Listen for class deletion indications.               |
| -di             | Listen for instance deletion indications.            |
| -mc             | Listen for class modification indications.           |
| -mi             | Listen for instance modification indications.        |
| -ri             | Listen for instance read indications.                |
| -pm             | Listen for instance pre-method indications.          |
| -tm             | Listen for instance post-method indications.         |
| -port           | Port to listen for indications on                    |
| -ssl            | Listen for indications using SSL                     |
| -ikf <file>     | Indication KeyStore File (Mutual Authentication)     |
| -ikp <password> | Indication KeyStore Password (Mutual Authentication) |
| -itf <file>     | Indication TrustStore File (Mutual Authentication)   |



# wbemind

## Example: Instance Creation

```
> wbemind -s https://localhost/interop -u root -p pokey -c Test_Test -ci
```

Listening for indications:

Hit "Enter" key at any time to quit.

\*\*\*\*\*

Start : Indication of Instance Creation

\*\*\*\*\*

```
instance of cim_instcreation {
    [EmbeddedObject(true)
    ]
    SourceInstance = instance of Test_Test {
        stringKey = "Brady";
        intKey = 12;
        stringProp = "Patriots";
        boolProp = null;
        arrayProp = {14, 2};
    };
};
```

```
};
```

\*\*\*\*\*

End : Indication of Instance Creation

\*\*\*\*\*

Hit "Enter" key at any time to quit.

Removing event subscriptions...

```
>
```



## **wbemextract Description**

Command Line utility that allows a user to generate MOF files containing the class, instance and qualifier definitions from the specified namespace.

useful for creating simulation environments, testing, comparison of changes,

Currently only supports CIM-XML

Future revision will support WS-Management



# wbemextract CLI Syntax

## wbemextract [-options]

### Usage Options:

|                  |  |
|------------------|--|
| -d               | Display debug information                        |
| -h               | Displays this usage statement                    |
| -kf <file>       | KeyStore File (Mutual Authentication)            |
| -kp <password>   | KeyStore Password (Mutual Authentication)        |
| -l <locales>     | List of locales in priority order.               |
| -o <dir>         | Directory to output files                        |
| -p <credential>  | WBEM Server credential (e.g. Password)           |
| -ph <host>       | HTTP Proxy Host                                  |
| -pp <host>       | HTTP Proxy Port                                  |
| -q               | Quiet mode                                       |
| -rp <credential> | WBEM Server Role credential (e.g. Password)      |
| -ru <principal>  | WBEM Server Role principal (e.g. Role to assume) |
| -s <server>      | WBEM Server URL                                  |
| -tf <file>       | TrustStore File (Mutual Authentication)          |
| -u <principal>   | WBEM Server principal (e.g. username)            |
| -v               | Displays the version information                 |
| -wp              | WBEM Protocol (e.g. CIM-XML, WS-Management)      |



# wbemextract Example:

```
> wbemextract -s https://localhost/interop -u root -p pokey -o tmp
```

```
Connecting to WBEM Server: https://localhost/interop
```

```
Searching namespace: /interop
```

```
Extracting Qualifier Types....
```

```
Extracting Classes....
```

```
Extracting Instances....
```

```
Extracted 42 qualifier types.
```

```
Extracted 100 classes.
```

```
Extracted 38 instances.
```

```
Extraction Completed!
```

```
> cd tmp
```

```
> dir /b *.mof
```

```
interop_instances.mof
```

```
interop_classes.mof
```

```
interop_qualifiers.mof
```

```
interop_schema.mof
```

```
> type interop_schema.mof
```

```
/* This file was generated by WBEMExtract from WBEM Solutions, Inc. */
```

```
#pragma namespace("/interop")
```

```
#pragma include("interop_qualifiers.mof")
```

```
#pragma include("interop_classes.mof")
```

```
#pragma include("interop_instances.mof")
```



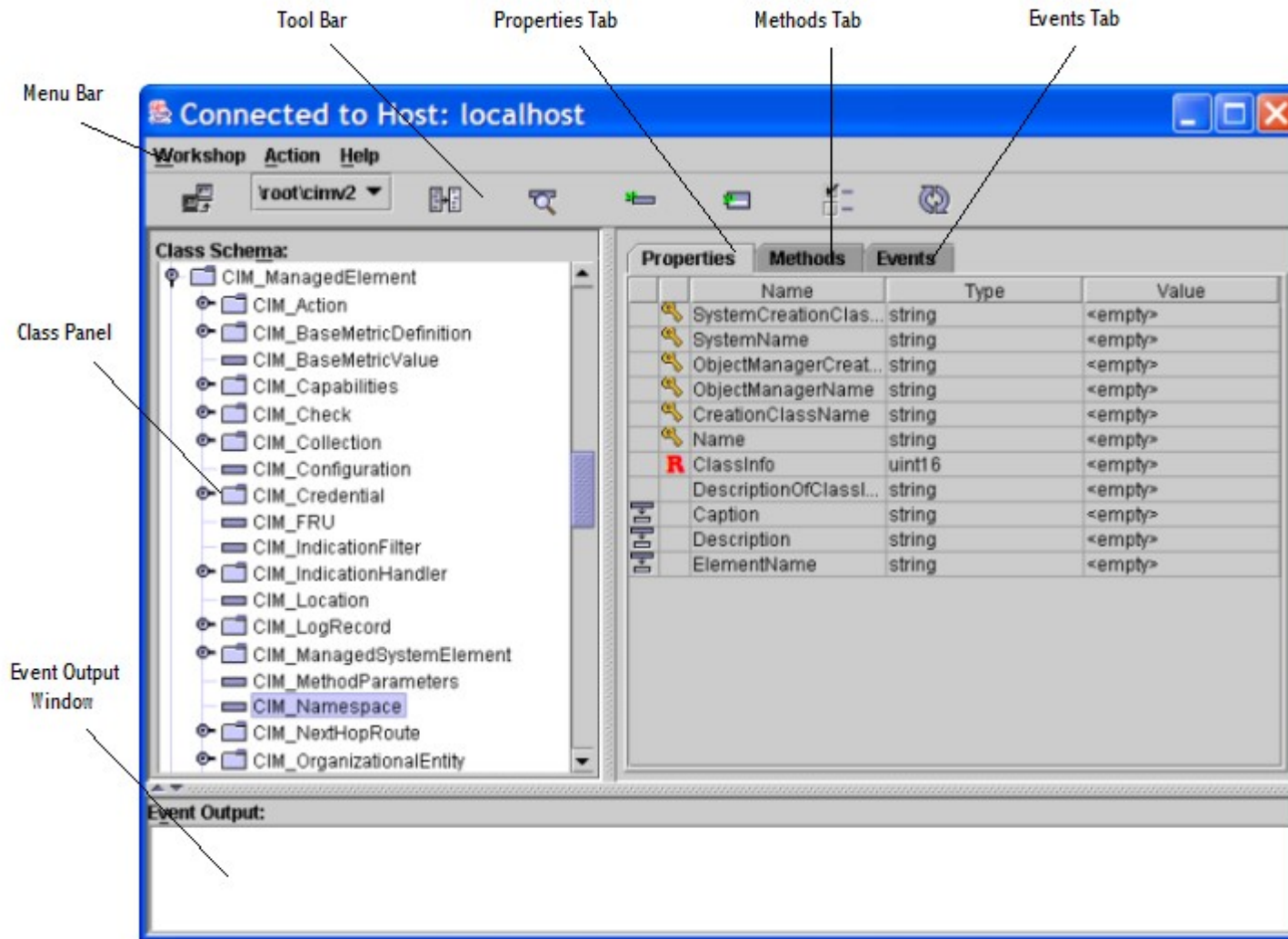
## **cimworkshop Description**

Graphical User Interface utility that allows a user to browse and manipulate CIM elements on a target WBEM Server.

Only supports CIM-XML

# cimworkshop

## Example:



The screenshot shows the cimworkshop application window titled "Connected to Host: localhost". The interface includes a menu bar with "Workshop", "Action", and "Help". Below the menu bar is a toolbar with icons for file operations and navigation. The main area is divided into two panes:

- Class Panel:** A tree view showing the class hierarchy. The selected class is "CIM\_Namespace".
- Properties Tab:** A table displaying the properties of the selected class. The table has columns for Name, Type, and Value.

Labels with arrows point to the following components:

- Menu Bar
- Tool Bar
- Properties Tab
- Methods Tab
- Events Tab
- Class Panel
- Event Output Window

| Name                  | Type   | Value   |
|-----------------------|--------|---------|
| SystemCreationClas... | string | <empty> |
| SystemName            | string | <empty> |
| ObjectManagerCreat... | string | <empty> |
| ObjectManagerName     | string | <empty> |
| CreationClassName     | string | <empty> |
| Name                  | string | <empty> |
| ClassInfo             | uint16 | <empty> |
| DescriptionOfClass... | string | <empty> |
| Caption               | string | <empty> |
| Description           | string | <empty> |
| ElementName           | string | <empty> |



# J WBEM Server



# J WBEM Server Features

WS-Management Clients

CIM-XML Clients and Indications

CMPI Provider support (beta)

JSR48 Provider

Use of Native OS Security Interface

IPv4, IPv6, HTTP, HTTPS

Mutual Authentication with SSL Certificates

Provider Code Authorization/Signing

Use behind a Firewall

Systems with multiple IP addresses

SLP Discovery

# WBEM Workshop





# WBEM Workshop

New tool

GUI (Graphical User Interface)

Available early next year

Replaces CIM Workshop

Included in SDK Pro

Modular “plugin” design

Provide support for all command line utilities



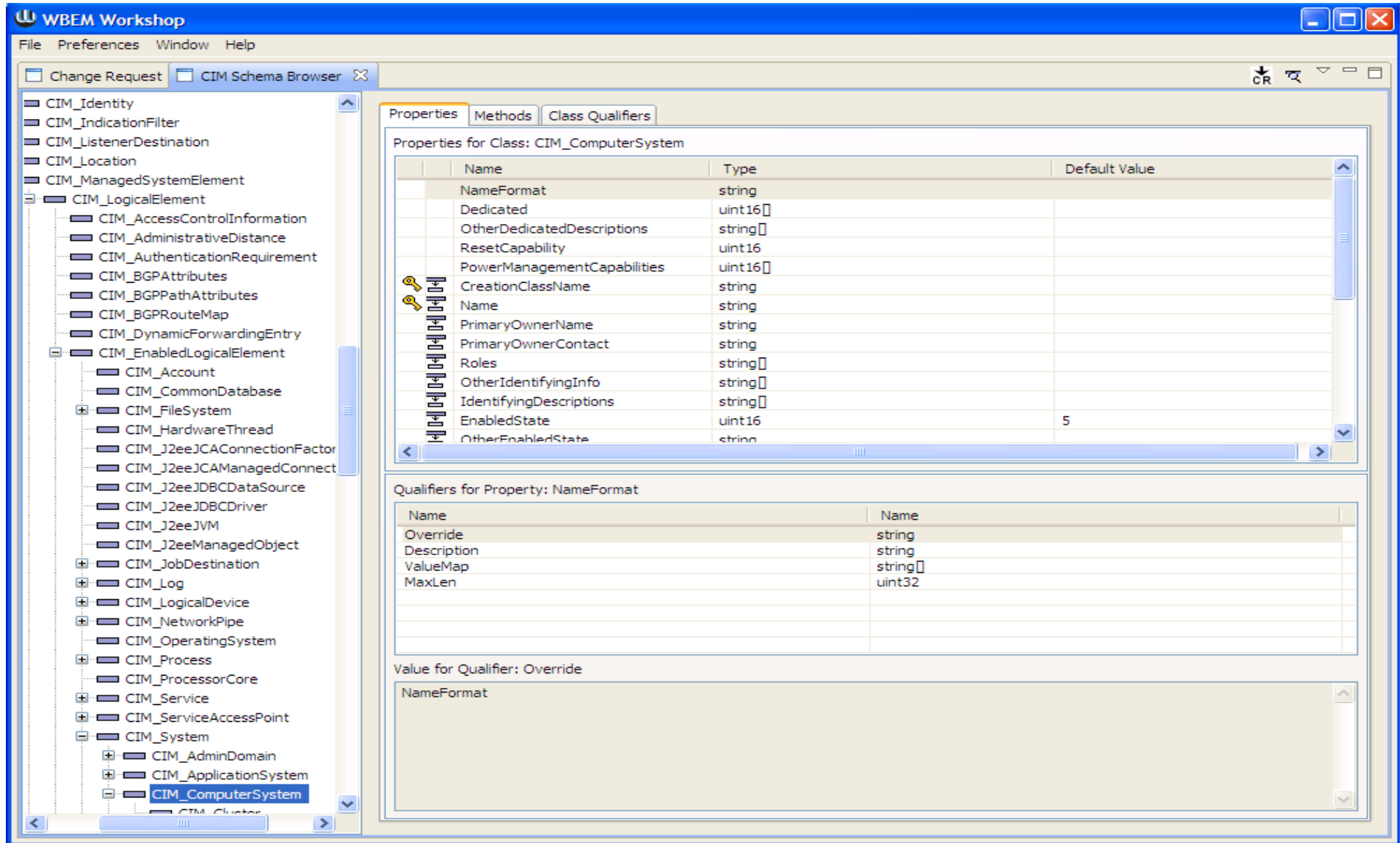
# WBEM Workshop Plug in

Over 20 “plugin” modules planned`

Some of the plugin modules available next year

- Schema Browser
- Instance Browser
- SLP
- WBEM Operations
- WBEM Indications
- Profile Browser
- Log Reader/Viewer

# Schema Browser



The screenshot shows the WBEM Workshop interface. The 'CIM Schema Browser' window is active, displaying a tree view of CIM classes on the left and a detailed view of the 'CIM\_ComputerSystem' class on the right.

**Properties for Class: CIM\_ComputerSystem**

| Name                        | Type     | Default Value |
|-----------------------------|----------|---------------|
| NameFormat                  | string   |               |
| Dedicated                   | uint16[] |               |
| OtherDedicatedDescriptions  | string[] |               |
| ResetCapability             | uint16   |               |
| PowerManagementCapabilities | uint16[] |               |
| CreationClassName           | string   |               |
| Name                        | string   |               |
| PrimaryOwnerName            | string   |               |
| PrimaryOwnerContact         | string   |               |
| Roles                       | string[] |               |
| OtherIdentifyingInfo        | string[] |               |
| IdentifyingDescriptions     | string[] |               |
| EnabledState                | uint16   | 5             |
| OtherEnabledState           | string   |               |

**Qualifiers for Property: NameFormat**

| Name        | Name     |
|-------------|----------|
| Override    | string   |
| Description | string   |
| ValueMap    | string[] |
| MaxLen      | uint32   |

Value for Qualifier: Override

NameFormat



Q & A

