

AIX 5L Version 5.2



# Common Information Model Guide



AIX 5L Version 5.2



# Common Information Model Guide

**Note**

Before using this information and the product it supports, read the information in Appendix H, "Notices," on page 45.

**First Edition (May 2004)**

This edition applies to AIX 5L Version 5.2 and to all subsequent releases of this product until otherwise indicated in new editions.

A reader's comment form is provided at the back of this publication. If the form has been removed, address comments to Information Development, Department H6DS-905-6C006, 11501 Burnet Road, Austin, Texas 78758-3493. To send comments electronically, use this commercial Internet address: aix6kpub@austin.ibm.com. Any information that you supply may be used without incurring any obligation to you.

Copyright 2004 Distributed Management Task Force, Inc. All rights reserved.

Copyright (c) 2000, 2001, 2002 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; The Open Group; Tivoli Systems. Copyright (c) 2003 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; EMC Corporation; The Open Group. Copyright (c) 2004 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; EMC Corporation; VERITAS Software Corporation; The Open Group.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE ABOVE COPYRIGHT NOTICE AND THIS PERMISSION NOTICE SHALL BE INCLUDED IN ALL COPIES OR SUBSTANTIAL PORTIONS OF THE SOFTWARE. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

© Copyright International Business Machines Corporation 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About This Book</b> . . . . .	v
Who Should Use This Book . . . . .	v
Highlighting . . . . .	v
Case-Sensitivity in AIX . . . . .	v
ISO 9000 . . . . .	v
<b>Chapter 1. Common Information Model overview</b> . . . . .	1
<b>Chapter 2. Install the Pegasus CIM Server and base providers</b> . . . . .	3
<b>Chapter 3. Configure the CIM Server</b> . . . . .	5
Install the OpenSSL RPM file . . . . .	5
Set resource limits . . . . .	5
Start and stop the CIM Server . . . . .	5
Add a user to the CIM Server . . . . .	6
Secure your CIM Server . . . . .	6
Set and configure CIM Server tracing . . . . .	7
Configure logging . . . . .	8
<b>Chapter 4. Recover and rebuild the repository</b> . . . . .	11
<b>Chapter 5. Base providers for AIX</b> . . . . .	13
MOF files . . . . .	13
Providers . . . . .	13
Set and configure AIX provider tracing . . . . .	25
<b>Appendix A. cimauth Command</b> . . . . .	27
Purpose . . . . .	27
Syntax . . . . .	27
Description . . . . .	27
Flags . . . . .	27
Return Values . . . . .	27
Examples . . . . .	28
Related Information . . . . .	28
<b>Appendix B. cimconfig Command</b> . . . . .	29
Purpose . . . . .	29
Syntax . . . . .	29
Description . . . . .	29
Flags . . . . .	29
Exit Status . . . . .	30
Examples . . . . .	30
<b>Appendix C. cimmo Command</b> . . . . .	31
Purpose . . . . .	31
Syntax . . . . .	31
Description . . . . .	31
Flags . . . . .	32
Exit Status . . . . .	32
Standard Error . . . . .	32
Examples . . . . .	32
Related Information . . . . .	33

<b>Appendix D. cimmoft Command</b>	35
Purpose	35
Syntax	35
Description	35
Flags	36
Exit Status	36
Examples	36
Standard Error	36
Related Information.	37
<b>Appendix E. cimprovider Command</b>	39
Purpose	39
Syntax	39
Description	39
Flags	39
Exit Status	40
Examples	40
Related Information.	40
<b>Appendix F. cimserver Command</b>	41
Purpose	41
Syntax	41
Description	41
Flags	41
Parameters.	41
Exit Status	41
Examples	41
Related Information.	41
<b>Appendix G. cimuser Command</b>	43
Purpose	43
Syntax	43
Description	43
Flags	43
Return Values.	44
Examples	44
Related Information.	44
<b>Appendix H. Notices</b>	45
Trademarks	46
<b>Index</b>	47

---

## About This Book

This book contains information for understanding the tasks required to use Common Information Model (CIM) instrumentation on AIX.

This edition supports the release of AIX 5L Version 5.2 with the 5200-03 Recommended Maintenance package. Any specific references to this maintenance package are indicated as *AIX 5.2 with 5200-03*.

---

## Who Should Use This Book

This book is for persons performing system management on the computer and operating system. Readers of this book are expected to know basic operating system commands.

It is assumed that you are familiar with the information and concepts presented in the following publications:

- *AIX 5L Version 5.2 System Management Guide: Operating System and Devices*
- *AIX 5L Version 5.2 Installation Guide and Reference*

---

## Highlighting

The following highlighting conventions are used in this book:

<b>Bold</b>	Identifies commands, subroutines, keywords, files, structures, directories, and other items whose names are predefined by the system. Also identifies graphical objects such as buttons, labels, and icons that the user selects.
<i>Italics</i>	Identifies parameters whose actual names or values are to be supplied by the user.
Monospace	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

---

## Case-Sensitivity in AIX

Everything in the AIX operating system is case-sensitive, which means that it distinguishes between uppercase and lowercase letters. For example, you can use the **ls** command to list files. If you type **LS**, the system responds that the command is "not found." Likewise, **FILEA**, **FiLea**, and **filea** are three distinct file names, even if they reside in the same directory. To avoid causing undesirable actions to be performed, always ensure that you use the correct case.

---

## ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.



---

## Chapter 1. Common Information Model overview

The Common Information Model (CIM) is a conceptual information model for describing management properties that is not bound to a particular implementation. This allows for the interchange of management information between management systems and applications through the Common Information Model Object Manager (CIMOM), which is an object management engine that exists between the managed system and the management application. Management applications can be local or remote, but currently CIMOM and the programs that collect management data, or *providers*, must be located on the machine that is being instrumented. Pegasus is one of the open source implementations of the CIMOM that adheres to the Distributed Management Task Force (DMTF) CIM and Web-based Enterprise Management (WBEM) standards. Pegasus is designed to be inherently portable and builds and runs on the AIX, Linux, and Windows operating systems.

The CIM Standard Schema provides the actual model descriptions. The CIM Schema supplies a set of classes with properties and associations that provide a conceptual framework within which it is possible to organize the available information about the managed environment. Platform-specific objects, such as AIX, that must be managed are defined as extensions to this standard CIM model.

Providers collect the management data from the underlying platform and supply it to the objects described in the conceptual CIM model, after which they can be served by the CIMOM to the management application through XML over HTTP. Management applications can monitor and change the properties of the underlying platform by sending a request to the CIMOM, which in turn would delegate it to the provider to monitor and change the request on the underlying platform. This mechanism provides an open-standard way for a management application to access and change management data for a particular platform.

CIM on AIX 5.2 includes the following:

- An open-source implementation of the CIMOM called Pegasus Version 2.3
- A CIM schema, version 2.7, that defines an information model for representing system management resources
- Providers that instrument a set of AIX resourced based on CIM schema, version 2.7

For additional information about Pegasus and WBEM, see the following:

- The OpenPegasus Web site at <http://www.openpegasus.org/>
- The DMTF Web site at <http://www.dmtf.org/standards/cim>
- The DMTF Web site at <http://www.dmtf.org/standards/wbem>
- The WBEM Web site at <http://www.wbemsolutions.com/tutorials/CIM/cim.html>



---

## Chapter 2. Install the Pegasus CIM Server and base providers

AIX 5.2 ships the following packages to support Pegasus:

- **sysmgt.pegasus.cimserver** installs the Pegasus CIM Server filesets in the **/opt/freeware/cimom/pegasus** directory
- **sysmgt.pegasus.osbaseproviders** installs the base providers for AIX filesets in the **/usr/pegasus/provider** directory

You can install the packages using either the System Management Interface Tool (SMIT) or the **installp** command.

For more information about using the **installp** command, see the **installp** command in *AIX 5L Version 5.2 Commands Reference, Volume 3*.

**Note:** Before continuing with the installation, review the license information.

To install the packages using SMIT, complete the following:

1. At the command line, type **smitty**.
2. Select **Software Installation and Maintenance>Install and Update Software>Install Software**.
3. At the Input Device/directory for software field, press the F4 key to view a list of options.
4. Select the option that reflects the location or media that contains the CIM packages.
5. At the Software to Install field, press the F4 key to view a list of package options.
6. Select the **sysmgt.pegasus.cimserver** and **sysmgt.pegasus.osbaseproviders** packages by pressing the F7 key.

To verify that the CIM Server filesets were installed correctly, use the **lslpp** command as follows:

```
lslpp -al sysmgt.pegasus.cimserver.rte
```

- If the installation completed successfully, a message similar to the following is returned:

```
lslpp -l sysmgt.pegasus.cimserver.rte
  Filesset              Level State      Description
-----
Path: /usr/lib/objrepos sysmgt.pegasus.cimserver.rte 2.3.1.0  COMMITTED \
Pegasus CIM Server Runtime Environment
```

- If the installation did not complete successfully, a message similar to the following is returned:

```
lslpp: Filesset sysmgt.pegasus.cimserver.rte not installed.
```

To verify that the base providers for AIX filesets were installed correctly, use the **lslpp** command as follows:

```
lslpp -al sysmgt.pegasus.osbaseproviders
```

- If the installation completed successfully, a message similar to the following is returned:

```
lslpp -l sysmgt.pegasus.osbaseproviders
  Filesset              Level State      Description
-----
Path: /usr/lib/objrepos sysmgt.pegasus.osbaseproviders 1.2.3.0  COMMITTED
Base Providers for AIX OS
```

- If the installation did not complete successfully, a message similar to the following is returned:

```
lslpp: Filesset sysmgt.pegasus.osbaseproviders not installed.
```



---

## Chapter 3. Configure the CIM Server

Your CIM Server requires limited configuration. After you install both fileset packages for the server and the providers, your Pegasus CIM Server is ready for use.

The AIX Pegasus CIM Server is SSL-ready and requires the **libssl.a** library to be available on the system for the CIM Server to run even if you decide not to use SSL. The **libssl.a** library is a part of the OpenSSL rpm file. For more information about enabling the CIM Server to run with SSL, see “Secure your CIM Server” on page 6.

---

### Install the OpenSSL RPM file

In order for the CIM Server to run, the OpenSSL rpm file must be installed. To determine if the rpm file is installed on your system, run the following commands:

```
rpm -q -f /opt/freeware/lib/libssl.a
rpm -qa | grep -i openssl
```

If both the libssl.a library and the openssl-0.9.6XXX rpm, where XXX indicates the build level, are found, then OpenSSL is installed on your system.

If OpenSSL is not already installed, you can find this rpm file on the *AIX Linux ToolBox* CD. It can also be downloaded from the AIX Toolbox for Linux Applications Web site at <http://www-1.ibm.com/servers/aix/products/aixos/linux/download.html>.

On this Web site, select **AIX Toolbox Cryptographic Content** under the Sorted Download heading on the right of the page. After you have registered and accepted the license, you can download the “openssl - Secure Sockets Layer and cryptography libraries and tools”, such as **openssl-0.9.6k-1.aix4.3.ppc.rpm**, or later version.

To install the OpenSSL rpm file, run the following command:

```
rpm -ivh openssl-0.9.6XXX.rpm
```

where XXX indicates the build level.

---

### Set resource limits

To maximize the CIM Server’s processing capacity, the root user’s **ulimit** for the Soft DATA segment must be set to -1, which is “unlimited”.

To set resource limits in SMIT, complete the following:

1. Type **smitty** user at a command prompt.
2. Select **Change/Show Characteristics of a User**.
3. Type **root** in the **User NAME** field.
4. Scroll to the **Soft DATA segment** field and change the value to -1.

In order for the change to take effect, you are required to log out and log back in again.

For more information about user and system resource limits, refer to the **ulimit** command in the Commands Reference.

---

### Start and stop the CIM Server

The CIM Server does not start or stop automatically on your system. To start or stop the CIM Server, you must run the following commands:

## **cimserver**

Starts the CIM Server

## **cimserver -s**

Stops the CIM Server

The CIM Server runs as a daemon in the background. By default, Basic Authentication is enabled. When Basic Authentication is enabled, the CIM Server authenticates the user ID and password of each request.

To disable Basic Authentication, set `enableAuthentication=false` in the configuration files before starting the CIM Server. If the CIM Server is already running, it must be restarted for the change to take effect. Client requests to the CIM Server must include a user ID and password if Basic Authentication is enabled.

---

## **Add a user to the CIM Server**

The CIM Server maintains the list of authorized users in the `/opt/freeware/cimom/pegasus/etc/cimserver.passwd` file. To add a user to this list, the user must be an existing AIX user. Run the following command to add a user to CIM Server:

```
cimuser -a -u id
```

where *id* is the user ID to be added.

The user will be prompted to enter a password to be associated with this *id*. This is a CIM Server password and does not have to be the same as the AIX Operating System login password.

For more information about the `cimuser` command, see Appendix G, “`cimuser` Command,” on page 43.

---

## **Secure your CIM Server**

To properly secure your CIM Server, it is recommended that you enable SSL when you are running the CIM Server. If you prefer to run your CIM Server in an unsecure mode, you can do so by either not enabling SSL, or disabling SSL after you enable it.

## **Create SSL certificates**

To enable your server to run in SSL mode, you must first create a certificate and two keys: one public key and one private key. Because the private key contains the public key information in OpenSSL, a separate public key is not required. For more information about public and private keys in OpenSSL, see the OpenSSL Web site at <http://www.openssl.org>.

To create a self-signed test certificate called `cert.pem` and an RSA private key called `file.pem`, run the following commands:

```
cd /opt/freeware/cimom/pegasus/etc
CN="Common Name"
EMAIL="test@email.address"
HOSTNAME=`hostname`
sed -e "s/$CN/$HOSTNAME/" -e "s/$EMAIL/root@$HOSTNAME/" orig/ssl.cnf > ssl1.cnf
/opt/freeware/bin/openssl req -x509 -days 365 -newkey rsa:512 -nodes \
-config ssl1.cnf -keyout file.pem -out cert.pem

cp cert.pem client.pem

rm ssl1.cnf
```

The `ssl.cnf` file is a sample configuration file for OpenSSL. Two fields in the file, `CN="Common Name"` and `EMAIL="test@email.address"`, are used to specify the identity to be certified. Use the `sed` command string to replace the placeholder information with your system's information. You can customize the common name and the e-mail address to match your environment.

Output from the **sed** command is directed to the temporary **ssl1.cnf** file.

The default location for the key and certificate files is the **/opt/freeware/cimom/pegasus/etc** directory. If another directory is used, it must be specified using the **cimconfig** command before the CIM Server is started.

For example, if you choose to store the certificate and key files in a directory called **/var/pegasus**, specify the location of these files using the following commands before starting the CIM Server:

- `cimconfig -p -s sslCertificateFilePath=/var/pegasus/cert.pem`
- `cimconfig -p -s sslKeyFilePath=/var/pegasus/file.pem`
- 

Use the commands in the preceding list to create the test certificate and private key. The trust file, **client.pem** is already created. This trust file is a copy of the default trust file, or the **client.pem** certificate file. If you use a different directory or file name, use the **PEGASUS\_SSLTRUSTFILEPATH** environment variable to specify the location of the trust file.

For example, if the trust file name and location is **/var/pegasus/trustfile.pem**, the following Korn shell command can be used to set the environment variable:

```
export PEGASUS_SSLTRUSTFILEPATH=/var/pegasus/trustfile.pem
```

## Enable the CIM Server with SSL

After the SSL certificates are created, enable the CIM Server with SSL by setting the following parameters to the specified values:

- `enableHttpsConnection=true`
- `enableHttpConnection=false`

These parameters can be set in one of the following places:

- the **cimconfig** command. For more information about the **cimconfig** command, see Appendix B, “cimconfig Command,” on page 29.
- command-line options to the **cimserver** command during the startup of the CIM Server. For more information about the **cimserver** command, see Appendix F, “cimserver Command,” on page 41.

---

## Set and configure CIM Server tracing

Tracing should only be used for debugging purposes and is, by default, disabled. You can, however, enable the tracing mechanism by specifying the trace level and the components that you want traced with the `traceLevel` configuration parameter. Choose from one of the following trace levels:

### level 1

Function entry and exit

### level 2

Basic flow trace messages, low data detail

### level 3

Interfunction logic flow, medium data detail

### level 4

All information, high data detail

Trace data is saved in the file specified by the `traceFilePath` configuration parameter. By default, the `traceFilePath` parameter is set to the **/opt/freeware/cimom/pegasus/logs/cimserver.trc** file.

To specify the component or components that you want to trace, use the `traceComponents` parameter. Some of the components that you can specify with this parameter include the following:

- Channel
- XmlWriter
- CimData
- ProvManager
- Authorization
- Authentication
- WQL
- Thread

You can also choose to trace all of the components by using the word ALL in place of a specific component name. If the *traceComponents* parameter is not set to any component, tracing is off regardless of the *traceLevel* parameter setting.

You can modify the trace configuration parameters by using one of the following methods:

- In the configuration files before the CIM Server is started. Use your favorite editor to change the values of the parameters.
- As command line options to the **cimserver** command during the startup of the CIM Server.
- As command line options to the **cimconfig** command while the CIM Server is running.

For example, to set the trace level to trace all information with high data detail in the **Thread** and **ProvManager** components, type the following commands:

```
cimconfig -s traceLevel=4
cimconfig -s traceComponents=Thread,ProvManager
```

Similarly, to disable all tracing, type the following command:

```
cimconfig -s traceComponents=
```

For more information about setting configuration parameters with the **cimconfig** command, see Appendix B, “cimconfig Command,” on page 29.

---

## Configure logging

The logging utility that is available for the Pegasus CIM Server is initially enabled and cannot be disabled. However, you can configure the utility by choosing the level of logging to use and specifying the directory in which to store the log files.

The following are the available levels of logging:

- TRACE
- INFORMATION
- SEVERE
- FATAL

You can change the logging level during the CIM startup by specifying the logging level with the **logLevel** parameter through one of the following methods:

- In the configuration files before the CIM Server is started
- As an option to the **cimserver** command during the CIM Server startup
- As an option to the **cimconfig** command while the CIM Server is running

For example, if you chose to use the **cimconfig** command method, you would type the following while the CIM Server is running:

```
cimconfig -s logLevel=INFORMATION
```

The log data is saved in the following files:

- **PegasusDebug.Log**
- **PegasusError.log**
- **PegasusStandard.log**
- **PegasusTrace.log**

These files are located in the directory that you specify with the **logdir** configuration parameter. By default, the **logdir** parameter is set to the **/opt/freeware/cimom/pegasus/logs** directory.



---

## Chapter 4. Recover and rebuild the repository

An active server repository is modified by the CIM Server when new schema classes are identified and loaded. By default, the CIM Repository is located at `/opt/freeware/cimom/pegasus/etc/repository`, and is pre-loaded with the CIM Schema 2.7, and the AIX extended classes of the CIM Schema 2.7.

An identical copy of the CIM Repository that is not used or modified by the CIM Server is also included. This repository copy exists in the `/opt/freeware/cimom/pegasus/etc/orig/repository` directory for recovery purposes only.

The server repository can be restored by using either of the following options:

- To recover your repository, delete the existing repository and copy the identical copy of the original repository by completing the following steps:

1. Verify that the CIM Server is not running. To stop the CIM Server, type the following command:

```
cimserver -s
```

For more information about stopping the CIM Server, see Appendix F, “cimserver Command,” on page 41.

2. Remove the active repository by typing the following command:

```
rm -rf /opt/freeware/cimom/pegasus/etc/repository
```

3. Copy the original repository by typing the following command:

```
cp -pr /opt/freeware/cimom/pegasus/etc/orig/repository /opt/freeware/cimom/pegasus/etc
```

4. After the repository is restored, start the CIM Server.

Your repository is now recovered.

**Note:** This method of recovery will restore the repository to its original state. To modify the repository to more accurately reflect any changes you made, use the **cimmof** or **cimmofl** commands. For more information about these commands, see Appendix C, “cimmof Command,” on page 31 and Appendix D, “cimmofl Command,” on page 35.

- The **rebuild\_repository** script available for recovering your repository rebuilds the CIM Repository by loading the Managed Object Format (MOF) files of CIM Schema 2.7 and AIX extended classes.

To use the **rebuild\_repository** script to recover your repository, complete the following steps:

1. Verify that the CIM Server is not running. To stop the CIM Server, type the following command:

```
cimserver -s
```

For more information about stopping the CIM Server, see Appendix F, “cimserver Command,” on page 41.

2. Run the **rebuild\_repository** script by typing the following command:

```
/opt/freeware/cimom/pegasus/etc/orig/rebuild_repository
```

Before the script continues, you will be asked to confirm the request to remove the active repository.

3. After the repository is recovered, start the CIM Server. For more information about starting the CIM Server, see Appendix F, “cimserver Command,” on page 41.

This restore option only restores the repository to the original state that was shipped with the package. Any additional classes that you added to the repository must be reloaded using the **cimmof** or **cimmofl** commands. For more information about these commands, see Appendix C, “cimmof Command,” on page 31 and Appendix D, “cimmofl Command,” on page 35.



---

## Chapter 5. Base providers for AIX

The Pegasus CIM Server uses providers to manage system resources. These providers are required to follow the implementation rules defined by the CIMOM. The Pegasus CIM Server supports its own proprietary C++ Provider API Interface, but also works with the Common Manageability Programming Interface (CMPI), a C Provider interface. Providers written to the CMPI provider interface are interoperable with the Pegasus CIM Server and other CIMOMs without any changes. For more information about CMPI, see the [www.openpegasus.org](http://www.openpegasus.org) Web site.

---

### MOF files

A **.mof** file is a text file that defines the attributes of a managed resource. An example of a managed resource is an operating system, which includes the following types of attributes, among others:

- Operating system type
- Version
- Date installed
- Number of users

The **.mof** file format adheres to the platform-neutral CIM standard of the DMTF industry consortium. The **mof** files for the IBM providers are located in the **/usr/pegasus/provider** directory.

The following **.mof** files are contained in the **/usr/pegasus/provider** directory:

#### **IBMAIX\_OSBase.mof**

Contains extensions to the standard CIM classes

#### **IBMAIX\_OSBaseRegistration.mof**

Registers the Base Providers for AIX with the Pegasus CIM Server

---

### Providers

The following provider types are shipped in the **sysmgt.pegasus.osbaseproviders** package and are supported by the Pegasus CIM Server:

#### **instance providers**

Provide a dynamic list of instances that are available in a CIMOM. For example, an Operating System Instance provider returns information about the instance of the Operating System that is running on the system.

#### **association providers**

Build associations between components dynamically. For example, the Pegasus CIM server contains information about the Computer System and the Operating System. An association provider associates the Computer System and the Operating System and returns information about this association.

The following CIM classes have been implemented by IBM-supplied providers to provide operating system information:

<b>CIM Class</b>	<b>IBM provider</b>
IBMAIX_AFS	OSBase_AFSPProvider
IBMAIX_BootOSFromFS	OSBase_BootOSFromFSPProvider
IBMAIX_CDFS	OSBase_CDFSPProvider
IBMAIX_CSNetworkPort	OSBase_CSNetworkPortProvider
IBMAIX_CSProcessor	OSBase_CSProcessorProvider

<b>CIM Class</b>	<b>IBM provider</b>
IBMAIX_ComputerSystem	OSBase_ComputerSystemProvider
IBMAIX_DFS	OSBase_DFSPProvider
IBMAIX_EthernetPort	OSBase_EthernetPortProvider
IBMAIX_FCPort	OSBase_FCPortProvider
IBMAIX_HostedFileSystem	OSBase_HostedFileSystemProvider
IBMAIX_IPProtocolEndpoint	OSBase_IPProtocolEndpointProvider
IBMAIX_JFS2	OSBase_JFS2Provider
IBMAIX_JFS	OSBase_JFSPProvider
IBMAIX_LocalLoopbackPort	OSBase_LoopBackPortProvider
IBMAIX_NFS	OSBase_NFSPProvider
IBMAIX_NetworkPortImplementsIPEndpoint	OSBase_NetworkPortImplementsEndpointProvider
IBMAIX_OSProcess	OSBase_OSProcessProvider
IBMAIX_OperatingSystem	OSBase_OperatingSystemProvider
IBMAIX_PROCFCS	OSBase_PROCFSPProvider
IBMAIX_Processor	OSBase_ProcessorProvider
IBMAIX_RunningOS	OSBase_RunningOSProvider
IBMAIX_TokenRingPort	OSBase_TokenRingPortProvider
IBMAIX_UnixProcess	OSBase_UnixProcessProvider

## OSBase\_AFSProvider

This provider is an instance provider that writes to the CMPI interface and provides information about remote AFS file systems.

The following provider properties are implemented:

<b>Property</b>	<b>Type</b>
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_AFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_BootOSFromFSProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between the current, running operating system and the file system from which the operating system is booted.

When an instance is returned, the properties of the IBMAIX\_OperatingSystem and subclasses of CIM\_FileSystem classes, listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_BootOSFromFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_CDFSPProvider

This provider is an instance provider that writes to the CMPI interface and provides information about local CDFS file systems.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_CDFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_CSNetworkPortProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between a computer system and its network ports.

When an instance is returned, the properties of the IBMAIX\_ComputerSystem and subclasses of CIM\_NetworkPort (Ethernet, TokenRing, FCPort and LoopBackPort) classes listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_CSNetworkPort class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_CSProcessorProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between a computer system and the processors that the computer system contains.

When an instance is returned, the properties of the IBMAIX\_ComputerSystem and subclasses of IBMAIX\_Processor classes, listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_CSProcessor class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_ComputerSystemProvider

This provider is an instance provider that writes to the CMPI interface and provides computer system properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
ElementName	string
Status	string
Name	string
CreationClassName	string
NameFormat	string
Dedicated[]	uint16A
OtherDedicatedDescriptions	string

For more information about the properties descriptions, see the IBMAIX\_ComputerSystem class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_DFSProvider

This provider is an instance provider that writes to the CMPI interface and provides information about remote DFS file systems.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_DFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_EthernetPortProvider

This provider is an instance provider that writes to the CMPI interface and provides Ethernet port properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
Element Name	string
Name	string
Status	string
EnabledState	uint16
OtherEnabledState	string
RequestedState	uint16
EnabledDefault	uint16
SystemCreationClassName	string
SystemName	string
CreationClassName	string
DeviceID	string
MaxSpeed	uint64 (bps)
Speed	uint64 (bps)
OtherNetworkPortType	string
LinkTechnology	uint16
OtherLinkTechnology	string

For more information about the properties descriptions, see the IBMAIX\_EthernetPort class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_FCPortProvider

This provider is an instance provider that writes to the CMPI interface and provides Fibre Channel port properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
Element Name	string
Name	string
Status	string
EnabledState	uint16
OtherEnabledState	string
RequestedState	uint16
EnabledDefault	uint16

Property	Type
SystemCreationClassName	string
SystemName	string
CreationClassName	string
DeviceID	string
MaxSpeed	uint64 (bps)
Speed	uint64 (bps)
OtherNetworkPortType	string
LinkTechnology	uint16
OtherLinkTechnology	string

For more information about the properties descriptions, see the IBMAIX\_FCPort class in the **/usr/pegasus/provider/mof/IBMAIX\_OSBase.mof** file.

## OSBase\_HostedFileSystemProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between the computer system container and its hosted local and remote file systems.

When an instance is returned, the properties of the IBMAIX\_ComputerSystem and subclasses of CIM\_FileSystem classes, listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_HostedFileSystem class in the **/usr/pegasus/provider/mof/IBMAIX\_OSBase.mof** file.

## OSBase\_IPProtocolEndpointProvider

This provider is an instance provider that writes to the CMPI interface and provides IP Protocol End Point properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
ElementName	string
Status	string
EnabledState	uint16
OtherEnabledState	string
RequestedState	uint16
EnabledDefault	uint16
CreationClassName	string
SystemCreationClassName	string
SystemName	string
CreationClassName	string
Name	string
NameFormat	string
Protocol Type	uint16

Property	Type
OtherTypeDescription	string
IPv4Address	string
IPv6Address	string
SubnetMask	string
PrefixLength	uint8
IPVersionSupport	uint16

For more information about the properties descriptions, see the `IBMAIX_IPProtocolEndpoint` class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_JFS2Provider

This provider is an instance provider that writes to the CMPI interface and provides information about local Enhanced Journaled File System.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the `IBMAIX_JFS2` class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_JFSProvider

This provider is an instance provider that writes to the CMPI interface and provides information about local Journaled File System.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string

Property	Type
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_JFS class in the **/usr/pegasus/provider/mof/IBMAIX\_OSBase.mof** file.

## OSBase\_LoopBackPortProvider

This provider is an instance provider that writes to the CMPI interface and provides Loop Back port properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
ElementName	string
Name	string
Status	string
EnabledState	uint16
OtherEnabledState	string
RequestedState	uint16
EnabledDefault	uint16
SystemCreationClassName	string
SystemName	string
CreationClassName	string
DeviceID	string
LinkTechnology	uint16
OtherLinkTechnology	string

For more information about the properties descriptions, see the IBMAIX\_LoopBackPort class in the **/usr/pegasus/provider/mof/IBMAIX\_OSBase.mof** file.

## OSBase\_NFSProvider

This provider is an instance provider that writes to the CMPI interface and provides information about remote NFS file systems.

The following provider properties are implemented:

Property	Type
Caption	string

Property	Type
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_NFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_NetworkPortImplementsEndpointProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between the IP Protocol End point and the network ports.

When an instance is returned, the properties of the IBMAIX\_IPProtocolEndpoint and subclasses of CIM\_NetworkPort (Ethernet, TokenRing, FCPort and LoopBackPort) classes listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_NetworkPortImplementsEndpoint class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_OSProcessProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between an operating system and the processes running in the operating system.

When an instance is returned, the properties of the IBMAIX\_OperatingSystem and subclasses of IBMAIX\_UnixProcess classes, listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_OSProcess class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_OperatingSystemProvider

This provider is an instance provider that writes to the CMPI interface and provides AIX operating system properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
Name	string
CSCreationClassName	string
CSName	string

Property	Type
CreationClassName	string
OSType	uint16
Version	string
LastBootUpTime	datetime
LocalDateTime	datetime
CurrentTimeZone	sint16
NumberOfProcesses	uint32
MaxNumberOfProcesses	uint32
TotalVirtualMemorySize	uint64
FreeVirtualMemory	uint64 (in KBytes)
FreePhysicalMemory	uint64 (in KBytes)
TotalVisibleMemorySize	uint64 (in KBytes)
SizedStoredInPagingFiles	uint64 (in KBytes)
FreeSpaceInPagingFiles	uint64 (in KBytes)
MaxProcessorsMemorySize	uint64 (in KBytes)
Distributed	Boolean
CodeSet	string
LanguageEdition	string
DefaultPageSize	uint32

For more information about the properties descriptions, see the `IBMAIX_OperatingSystem` class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_PROCFSPProvider

This provider is an instance provider that writes to the CMPI interface and provides information about local PROCFS file system.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
CSCreationClassName	string
CSName	string
CreationClassName	string
Name	string
Root	string
FilesystemSize	uint64
AvailableSpace	uint64
ReadOnly	Boolean
FilesystemType	string

For more information about the properties descriptions, see the IBMAIX\_PROCFS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_ProcessorProvider

This provider is an instance provider that writes to the CMPI interface and enumerates all processes in the computer system and provides individual process properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
ElementName	string
SystemCreationClassName	string
SystemName	string
CreationClassName	string
DeviceID	string
Role	string
Family	uint16
OtherFamilyDescription	string
MaxClockSpeed	uint32 (Mhz)
CurrentClockSpeed	uint32 (Mhz)
LoadPercentage	uint16
Stepping	string
CPU Status	uint16

For more information about the properties descriptions, see the IBMAIX\_UnixProcess class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase\_RunningOSProvider

This provider is an association and instance provider that writes to the CMPI interface and provides for the association between the computer system and the operating system.

When an instance is returned, the properties of the IBMAIX\_ComputerSystem and subclasses of IBMAIX\_OperatingSystem classes, listed in their respective providers, are returned.

For more information about the properties descriptions, see the IBMAIX\_RunningOS class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

## OSBase-TokenRingPortProvider

This provider is an instance provider that writes to the CMPI interface and provides token-ring port properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string

Property	Type
Element Name	string
Name	string
Status	string
EnabledState	uint16
OtherEnabledState	string
RequestedState	uint16
EnabledDefault	uint16
SystemCreationClassName	string
SystemName	string
CreationClassName	string
DeviceID	string
MaxSpeed	uint64 (bps)
Speed	uint64 (bps)
OtherNetworkPortType	string
LinkTechnology	uint16
OtherLinkTechnology	string

For more information about the properties descriptions, see the IBMAIX\_TokenRingPort class in the **/usr/pegasus/provider/mof/IBMAIX\_OSBase.mof** file.

## OSBase\_UnixProcessProvider

This provider is an instance provider that writes to the CMPI interface and enumerates all processes running in the system and provides individual process properties.

The following provider properties are implemented:

Property	Type
Caption	string
Description	string
Name	string
CSCreationClassName	string
CSName	string
CreationClassName	string
OSCreationClassName	string
OSName	string
Handle	string
CreationDate	datetime
KernelModeTime	uint64
UserModeTime	uint64
ParentProcessID	string
RealUserID	uint64
ProcessGroupID	uint64

Property	Type
ProcessTTY	string
ModulePath	string
Parameters[]	string

For more information about the properties descriptions, see the `IBMAIX_UnixProcess` class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

---

## Set and configure AIX provider tracing

Tracing should only be used for debugging purposes and is, by default, disabled. You can, however, enable the tracing utility for providers by setting the following environment variables before starting the CIM Server:

**Note:** You must use the appropriate shell command to set these variables. For example, for a Bourne shell or Korn shell, use the **export** command, and for the C shell, use the **setenv** command.

- **AIX\_PROVIDER\_TRACE\_FILE**=*filepath*, where *filepath* is the file where you want trace data saved.
- **AIX\_PROVIDER\_TRACE\_LEVEL**=*level*, where *level* is one of the trace levels.

Choose from one of the following trace levels:

### level 1

Function entry and exit

### level 2

Basic flow trace messages, low data detail

### level 3

Interfunction logic flow, medium data detail

### level 4

All information, high data detail

For example, to set the trace level to trace all information with high data detail using a Korn shell, type the following commands:

```
export AIX_PROVIDER_TRACE_FILE=/tmp/FILE_NAME
export AIX_PROVIDER_TRACE_LEVEL=4
```



---

## Appendix A. cimauth Command

---

### Purpose

Adds, modifies, removes or lists CIM user authorizations.

---

### Syntax

**cimauth -a -u *username* -n *namespace* [ -R ] [ -W ]**

**cimauth -m -u *username* -n *namespace* [ -R ] [ -W ]**

**cimauth -r -u *username* [ -n *namespace* ]**

**cimauth -l**

---

### Description

The **cimauth** command provides a command line interface to manage CIM user authorizations on a namespace. A namespace is a logical unit for grouping classes and instances to control their scope and visibility. These namespaces are not physical locations, but are similar to logical databases containing specific classes and instances.

This command does not configure or list CIM user password information. For more information on managing users, see the **cimuser** command.

The **cimauth** command can be used to remove authorizations of one user on one namespace or all the namespaces on which the user has authorizations. If no namespace is specified, then authorizations on all the namespaces for the specified user will be removed.

Specifying no options with the **cimauth** command will show the usage of the command.

---

### Flags

- a	Adds authorizations for a user on a namespace.
- l	Displays the authorizations of all CIM users.
- m	Modifies the authorizations for a user on a namespace.
- n <i>namespace</i>	Specifies the namespace for which you are adding authorizations to users.
- r	Removes the authorizations for a user on a namespace.
- R	Grants read authorization.
- u <i>username</i>	Specifies the user name for which you are adding, deleting, or modifying authorizations.
- W	Grants write authorization.

---

### Return Values

0	The command completed successfully.
1	The command did not complete successfully. An explanatory error message is written to stderr.

---

---

## Examples

1. To add read-write authorization to the user named **guest** on the **root/system** namespace, type the following:

```
cimauth -a -u guest -n root/system -R -W
```

2. To add read authorization to the user named **guest** on the **root/cimv2** namespace, type the following:

```
cimauth -a -u guest -n root/cimv2
```

3. To add read access only to the user named **guest** on the **root/system** namespace, type the following:

```
cimauth -m -u guest -n root/system -R
```

4. To remove all access granted to the user named **guest** on the **root/system** namespace, type the following:

```
cimauth -r -u guest -n root/system
```

5. To display the list of authorized user names, namespaces and authorizations, type the following:

```
cimauth -l
```

---

## Related Information

The cimuser command.

---

## Appendix B. cimconfig Command

---

### Purpose

Obtains, sets, or lists CIM Server configuration properties.

---

### Syntax

**cimconfig -g** *name* [ **-c** ] [ **-p** ] [ **-d** ]

**cimconfig -s** *name=value* [ **-c** ] [ **-p** ]

**cimconfig -u** *name* [ **-c** ] [ **-p** ]

**cimconfig -l** [ **-c** | **-p** ]

---

### Description

The **cimconfig** command provides a command line interface to manage CIM Server configuration properties. An error message is returned if the CIM Server is not running. Specifying no flags with the **cimconfig** command displays the command's usage statement.

---

### Flags

<b>-c</b>	Indicates that the action taken applies to the current configuration property value. This flag must be used in conjunction with one of the <b>-g</b> , <b>-s</b> , <b>-u</b> , or <b>-l</b> flags.
<b>-d</b>	Indicates that the action taken applies to the default configuration property value. This flag must be used in conjunction with the <b>-g</b> flag.
<b>-g</b> <i>name</i>	Displays the current, planned, or default value of the specified configuration property depending on whether the <b>-c</b> , <b>-p</b> , or <b>-d</b> flag is specified. If no other flag is specified, the current value of the specified configuration property is displayed.
<b>-l</b>	Lists the names of all the configuration properties when used without any other flag. When this flag is paired with the <b>-c</b> flag, a paired listing of all the current configuration properties and their values is displayed. When this flag is paired with the <b>-p</b> flag, a paired listing of all the planned configuration properties and their values is displayed.
<b>-p</b>	Indicates that the action taken applies to the planned configuration property value. This flag must be used in conjunction with one of the <b>-g</b> , <b>-s</b> , <b>-u</b> , or <b>-l</b> flags.
<b>-s</b> <i>name=value</i>	Sets the current or planned value of the specified configuration property to the specified value, depending on whether the <b>-c</b> or <b>-p</b> flag is specified. If no other flag is specified, the current value of the specified configuration property is set to the specified value. The <i>name</i> parameter is the name of the configuration property, and <i>value</i> is the new value for the configuration property. An error message is returned if the specified property cannot be set dynamically.

**-u** *name*

Resets the current or planned value of the specified configuration property to the default value depending on whether the **-c** or **-p** flag is specified. If no other flag is specified, the current value of the specified configuration property is reset to the default value. An error message is returned if the specified property cannot be updated dynamically.

---

## Exit Status

0	The command completed successfully.
1	The command did not complete successfully, and an explanatory message is written to stderr.

---

## Examples

1. To view the current value for the **port** configuration property, type one of the following commands:

```
cimconfig -g port
cimconfig -g port -c
```

2. To view the planned value of the **traceLevel** configuration property, type the following:

```
cimconfig -g traceLevel -p
```

3. To change the current value of the **traceLevel** configuration property to the value 2, type one of the following commands:

```
cimconfig -s traceLevel=2
cimconfig -s traceLevel=2 -c
```

4. To change the planned value of the **traceLevel** property to a value of 3, type the following:

```
cimconfig -s traceLevel=3 -p
```

5. To reset the current value of the **traceLevel** property to the default value, type one of the following commands:

```
cimconfig -u traceLevel
cimconfig -u traceLevel -c
```

6. To reset the planned value of the **traceLevel** property to the default value, type the following:

```
cimconfig -u traceLevel -p
```

7. To list all of the current configuration properties and their values, type the following:

```
cimconfig -l -c
```

---

## Appendix C. cimmoF Command

---

### Purpose

Compiles MOF files into the CIM repository through the CIM server.

---

### Syntax

**cimmoF -h**

**cimmoF** [ **-w** ] [ **-l** *path*] [ **-n** *namespace*] *file...file*

---

### Description

The **cimmoF** command is the command line interface to the Managed Object Format (MOF) Compiler. The MOF Compiler is a utility that compiles MOF files (using the MOF format defined by the DMTF CIM Specification) into CIM classes and instances that are stored in the CIM Repository.

#### Notes:

1. A superuser or user with write access to the default or specified namespace is the only user who can run the **cimmoF** command to compile MOFs in the CIM Repository.
2. Superclasses must be compiled before subclasses. Otherwise, the compile will fail.
3. It is strongly recommended that MOF files include all necessary subclasses, so they can compile properly even if certain classes are not in the CIM Repository.

The **cimmoF** command can be used to compile MOF files at any time after installation. If no input file is specified, stdin is used as the input.

The MOF Compiler requires that the input MOF files be in the current directory or that a fully qualified path be given. To simplify the specification of multiple MOF files in the **cimmoF** command line, the MOF Compiler allows compiling from files containing a list of MOF files using the **include** pragma, similar to the following:

- #pragma include ("application.mof")
- #pragma include ("server.mof")

MOF files using the include pragma must be in the current directory or in a directory specified by the **-l** flag.

The **-n namespace** flag can be used to specify a namespace in which the CIM classes and instances will be compiled. A namespace is a logical unit for grouping classes and instances to control their scope and visibility. These namespaces are not physical locations, but are similar to logical databases containing specific classes and instances. If this flag is not specified, the default namespace is **root/cimv2**, except for the provider registration schemas.

For provider registration schemas, if the **-n** flag is not specified, the default namespace is **root/PG\_InterOp**. If the **-n** flag is specified, the namespace specified must be **root/PG\_InterOp**, otherwise, the following error message is returned:

The requested operation is not supported.

For provider MOF files, the namespace specified must match one of the namespaces specified in the **PG\_ProviderCapabilities** class schema definition.

---

## Flags

<b>-h</b>	Displays the command usage and the version number of the MOF compiler.
<b>-I path</b>	Specifies the path to the included MOF files. This path can be relative or absolute.  If the input MOF file has #pragma includes, and the included files do not reside in the current directory, this option must be used to specify a path to them.
<b>-n namespace</b>	Specifies the namespace and overrides the default CIM Repository namespace. Any namespace specified here must be a valid CIM namespace.  For provider registration schemas, specify the <b>root/PG_InterOp</b> namespace.
<b>-w</b>	Suppresses warning messages. When MOF files are compiling, if there are CIM elements such as classes or instances defined in the MOF files that already exist in the CIM Repository, the <b>cimmof</b> command returns warning messages. Use this flag to suppress those warning messages.

---

## Exit Status

<b>0</b>	The command completed successfully.
<b>1</b>	The command did not complete successfully, and an error message is written to stderr.

---

## Standard Error

Error	Description
Error trying to create Repository in path localhost:5988: Cannot connect to: localhost:5988 Failed to set DefaultNamespacePath. Line number	The CIM Server is not running. Start the CIM Server and run the <b>cimmof</b> command again.
Operation cannot be carried out since the specified superclass does not exist.	An error was found at the line number line in the MOF file that the MOF compiler is parsing. The MOF compiler compiled a MOF file with one or more superclasses that are not in the CIM Repository.

---

## Examples

- To compile an MOF file named **processinfo.mof** into the default namespace in the CIM Repository, type the following:  

```
cimmof processinfo.mof
```
- To compile the MOF file name **processinfo.mof** into the **root/application** namespace, type the following:  

```
cimmof -n root/application processinfo.mof
```
- To compile the **CIMSchema25.mof** file defined in the **./MOF** directory that also contains #pragma includes for other MOF files in the **./MOF** directory, type the following:  

```
cimmof -w -I ./MOF MOF/CIMSchema25.mof
```

4. To list the arguments to the **cimmof** command and display the version of the MOF Compiler, type the following:

```
cimmof -h
```

---

## Related Information

The cimserver command.



---

## Appendix D. cimmoft Command

---

### Purpose

Compiles MOF files into the CIM repository. Run this command while the CIM Server is not running.

---

### Syntax

**cimmoft -h**

**cimmoft** [ **-w** ] [ **-I path** ] [ **-n namespace** ] *file...file*

---

### Description

The **cimmoft** command is the command line interface to the Managed Object Format (MOF) Compiler. The MOF Compiler is a utility that compiles MOF files (using the MOF format defined by the DMTF CIM Specification) into CIM classes and instances that are stored in the CIM Repository. To see changes that are made to the repository, you must restart the CIM server.

#### Notes:

1. A superuser or user with write access to the default or specified namespace is the only user who can run the **cimmoft** command to compile MOFs in the CIM Repository.
2. Superclasses must be compiled before subclasses. Otherwise, the compile will fail.
3. It is strongly recommended that MOF files include all necessary subclasses, so they can compile properly even if certain classes are not in the CIM Repository.

The **cimmoft** command can be used to compile MOF files at any time after installation. If no input file is specified, stdin is used as the input.

The MOF Compiler requires that the input MOF files be in the current directory or that a fully qualified path be given. To simplify the specification of multiple MOF files in the **cimmoft** command line, the MOF Compiler allows compiling from files containing a list of MOF files using the **include** pragma, similar to the following:

- #pragma include ("application.mof")
- #pragma include ("server.mof")

MOF files using the include pragma must be in the current directory or in a directory specified by the **-I** flag.

The **-n** flag can be used to specify a namespace in which the CIM classes and instances will be compiled. A namespace is a logical unit for grouping classes and instances to control their scope and visibility. These namespaces are not physical locations, but are similar to logical databases containing specific classes and instances. If this flag is not specified, the default namespace is **root/cimv2**, except for the provider registration schemas.

For provider registration schemas, if the **-n** flag is not specified, the default namespace is **root/PG\_InterOp**. If the **-n** flag is specified, the namespace specified must be **root/PG\_InterOp**, otherwise, the following error message is returned:

The requested operation is not supported.

For provider MOF files, the namespace specified must match one of the namespaces specified in the **PG\_ProviderCapabilities** class schema definition.

---

## Flags

<b>-h</b>	Displays the command usage and the version number of the MOF compiler.
<b>-I path</b>	Specifies the path to the included MOF files. This path can be relative or absolute.  If the input MOF file has #pragma includes, and the included files do not reside in the current directory, this option must be used to specify a path to them.
<b>-n namespace</b>	Specifies the namespace and overrides the default CIM Repository namespace. Any namespace specified here must be a valid CIM namespace.  For provider registration schemas, specify the <b>root/PG_InterOp</b> namespace.
<b>-w</b>	Suppresses warning messages. When MOF files are compiling, if there are CIM elements such as classes or instances defined in the MOF files that already exist in the CIM Repository, the <b>cimmofl</b> command returns warning messages. Use this flag to suppress those warning messages.

---

## Exit Status

<b>0</b>	The command completed successfully.
<b>1</b>	The command did not complete successfully, and an error message is written to stderr.

---

## Examples

- To compile an MOF file named **processinfo.mof** into the default namespace in the CIM Repository, type the following:  

```
cimmofl processinfo.mof
```
- To compile the **processinfo.mof** file into the **root/application** namespace, type the following:  

```
cimmofl -n root/application processinfo.mof
```
- To compile the **CIMSchema25.mof** file defined in the **./MOF** directory that also contains #pragma includes for other MOF files in the **./MOF** directory, type the following:  

```
cimmofl -w -I ./MOF MOF/CIMSchema25.mof
```
- To list the arguments to the **cimmofl** command and display the version of the MOF Compiler, type the following:  

```
cimmofl -h
```

---

## Standard Error

### Error

**Error trying to create Repository in path localhost:5988 Cannot connect to: localhost:5988 Failed to set DefaultNamespacePath.**

*Line number*

**Operation cannot be carried out since the specified superclass does not exist.**

### Description

The CIM Server is not running. Start the CIM Server and run the **cimmofl** command again.

An error was found at the *line number* line in the MOF file that the MOF compiler is parsing.

The MOF compiler compiled an MOF file with one or more superclasses that are not in the CIM Repository.

---

## Related Information

The cimserver command.



---

## Appendix E. cimprovider Command

---

### Purpose

Disables, enables, removes and lists registered CIM providers or CIM provider modules and module status

---

### Syntax

**cimprovider -d -m** *module*

**cimprovider -e -m** *module*

**cimprovider -r -m** *module* [-p *provider*]

**cimprovider -l** [-s | -m *module*]

---

### Description

The **cimprovider** command provides a command line interface to disable, enable, unregister, and list registered CIM providers. If a CIM provider is disabled, the CIM server rejects any requests to the provider. If a CIM provider is enabled, the CIM server forwards requests to the provider. If a CIM provider is unregistered, the CIM server will no longer have information about the provider.

In order to use the **cimprovider** command, the CIM Server has to be running and the specified provider or provider module, which is a grouping of providers in the same shared library, must be registered with Web-based Enterprise Management (WBEM) Services.

To list all providers in all modules, issue a **cimprovider -l** command, followed by the **cimprovider -l -m** for each listed module.

Specifying no options with the **cimprovider** command displays the command usage.

**Note:** The **-l** flag is available to any user, however, all other flags are available only to superusers.

---

### Flags

<b>-d</b>	Disables the specified CIM provider module, placing all of its contained providers in the Stopped state. When a specified provider module is in the disabled state, any new requests to its contained providers are rejected.  If a user attempts to disable a module that is already disabled, an error message is returned and no action is taken.
<b>-e</b>	Enables the specified CIM provider module, placing all of its contained providers in the OK state. Providers contained in the enabled provider module are available to accept new requests.  If a user attempts to enable a module that is already enabled or attempts to enable a module that is disabling, an error message is returned and no action is taken.

<b>-l</b>	Displays all the registered provider modules when it is not used with the <b>-m module</b> option.
<b>-m module</b>	Displays all the providers in the specified provider module when used with the <b>-m module</b> option.
<b>-p provider</b>	Specifies the provider module for the operation.
<b>-r</b>	Specifies the provider for the operation.
	Removes, or un-registers, the specified provider module and all of its contained providers. If a provider is specified, only that provider is removed. Other providers in the same provider module are not affected.
	To re-register a provider or provider module, the registration schema for that provider or provider module must be reloaded using the <b>cimmof</b> command.
<b>-s</b>	Displays the status of provider modules.

---

## Exit Status

<b>0</b>	The command completed successfully.
<b>1</b>	The command did not complete successfully, and an explanatory message is written to stderr.

---

## Examples

- To disable the **OperatingSystemProvider** provider module and all of its contained providers, type the following:  

```
cimprovider -d -m OperatingSystemProvider
```
- To enable the **OperatingSystemProvider** provider module and all of its contained providers, type the following:  

```
cimprovider -e -m OperatingSystemProvider
```
- To remove the **OperatingSystemProvider** provider module and all of its contained providers, type the following:  

```
cimprovider -r -m OperatingSystemProvider
```
- To remove the **PG\_OperatingSystemProvider** provider that is contained within the **OperatingSystemProvider** provider, type the following:  

```
cimprovider -r -m OperatingSystemProvider -p PG_OperatingSystemProvider
```
- To list the registered provider modules, type the following:  

```
cimprovider -l
```
- To list the registered provider modules and their status, type the following:  

```
cimprovider -l -s
```
- To list the registered providers contained within the **OperatingSystemProvider** provider module, type the following:  

```
cimprovider -l -m OperatingSystemProvider
```
- 

---

## Related Information

The **cimmof** and **cimserver** commands.

---

## Appendix F. cimserver Command

---

### Purpose

Start and stop the CIM Server.

---

### Syntax

`cimserver [[-v | -h | -s ] | [configProperty=value . . . ] ]`

---

### Description

The `cimserver` command allows you to start and stop the CIM Server. You can also set configuration properties and values with this command, overriding existing or default properties and values in the `cimserver_current.conf` and `cimserver_planned.conf` configuration files.

---

### Flags

<code>-h</code>	Prints the usage statement for this command.
<code>-s</code>	Stops the CIM Server.
<code>-v</code>	Displays the CIM Server version number.

---

### Parameters

<code>configProperty=value</code>	Starts the CIM Server with the <code>configProperty</code> set to <code>value</code> . Separate multiple <code>configProperty=value</code> pairs with a blank space. This command line option overrides existing or default values in the <code>cimserver_current.conf</code> and <code>cimserver_planned.conf</code> configuration files.
-----------------------------------	--

---

### Exit Status

<code>0</code>	The command completed successfully.
<code>&gt;0</code>	The command did not complete successfully, and an explanatory message is written to stderr.

---

### Examples

1. To start the CIM Server with a trace of level 4 on all components, type the following:

```
cimserver traceLevel=4 traceComponent=ALL
```

---

### Related Information

The `cimconfig` command.



---

## Appendix G. cimuser Command

---

### Purpose

Adds, modifies, removes or lists authorized users of the Pegasus CIM Server.

---

### Syntax

**cimuser -a -u** *username* [ **-w** *password* ]

**cimuser -m -u** *username* [ **-w** *password* ] [ **-n** *newpassword* ]

**cimuser -r -u** *username*

**cimuser -l**

---

### Description

The **cimuser** command allows you to manage CIM users by adding, deleting, listing, and modifying the passwords of users you authorize to use the Pegasus CIM Server.

The default location of the CIM password file is **/opt/freeware/cimom/pegasus/etc/cimserver.passwd**, but can be changed by setting the `passwordFilePath` property using the **cimconfig** command. For more information about **cimconfig**, see Appendix B, “cimconfig Command,” on page 29.

If you are completing a task that requires a password and you do not specify the password with the **-w password** flag, you will be prompted to enter the password. This password must be no more than 8 characters. If more than 8 characters are entered, only the first 8 characters will be recognized as the password.

When adding a new CIM user, the CIM user must be a valid user on the local system. There is no default authorization permissions set for newly added users. For more information on adding authorization information, see Appendix A, “cimauth Command,” on page 27.

Specifying no options with the **cimuser** command will display the usage statement.

---

### Flags

<b>- a</b>	Adds a CIM user.
<b>- l</b>	Lists all current CIM users.
<b>- m</b>	Modifies the password for the user specified in the <b>- u</b> <i>username</i> option.
<b>- n</b> <i>newpassword</i>	Identifies the new password for the user specified in the <b>- u</b> <i>username</i> option.
<b>- r</b>	Removes a CIM user. Any authorizations for the specified user will be deleted.
<b>- u</b> <i>username</i>	Identifies the user you want to add, remove, or for whom you want to modify the password.
<b>- w</b> <i>password</i>	Identifies the password for the user specified in the <b>- u</b> <i>username</i> option.

---

## Return Values

The **cimuser** command returns one of the following values:

0	The command completed successfully.
1	An error occurred. An explanatory message is written to stderr.

---

## Examples

1. To add a new user named **guest** with a password of **guest**, type the following:

```
cimuser -a -u guest -w guest
```

2. To add a new user named **guest** without specifying the password in the command, type the following:

```
cimuser -a -u guest
Please enter your password:
Please re-enter your password:
```

Because you did not specify the password in the initial command string with the **-w password** option, you are prompted to enter and re-enter the password.

3. To change the password for user **guest** to **bar**, type the following:

```
cimuser -m -u guest -w guest -n bar
```

4. To remove the user named **guest**, type the following:

```
cimuser -r -u guest
```

---

## Related Information

The **cimauth** command, the **cimconfig** command.

---

## Appendix H. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Dept. LRAS/Bldg. 003  
11400 Burnet Road  
Austin, TX 78758-3498  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX  
AIX 5L  
IBM  
RS/6000

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be the trademarks or service marks of others.

---

# Index

## C

- cimauth command 27
- cimconfig command 29
- cimmof command 31
- cimmofl command 35
- CIMOM i
- cimprovider command 39
- cimserver command 41
- cimuser command 43
- commands
  - cimauth 27
  - cimconfig 29
  - cimmof 31
  - cimmofl 35
  - cimprovider 39
  - cimserver 41
  - cimuser 43
  - sed 6
- configuration files 5
- configure
  - CIM Server 5
  - http port 5
  - https port 5
  - SSL mode 5

## D

- DMTF i

## I

- install
  - filesets 3
  - installp command 3
  - procedure 3

## M

- management applications i
- management systems i

## S

- schema i
- secure server 6
- sed command 6
- SSL
  - certificates 6
  - enable 6
  - keys 6
  - mode 6
  - public key 6
  - RSA key 6

## W

- WBEM i



---

# Readers' Comments — We'd Like to Hear from You

AIX 5L Version 5.2  
Common Information Model Guide

Publication No. SC23-4879-00

**Overall, how satisfied are you with the information in this book?**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

**How satisfied are you that the information in this book is:**

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



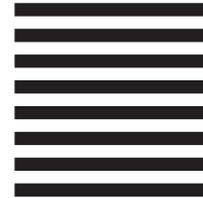
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Information Development  
Department H6DS-905-6C006  
11501 Burnet Road  
Austin, TX 78758-3493



Fold and Tape

Please do not staple

Fold and Tape





Printed in U.S.A.

SC23-4879-00

