Common Information Model Guide
Contents

About this book ................................................................. v
Highlighting ................................................................. v
Case-sensitivity in AIX .................................................... v
ISO 9000 ................................................................. v

Chapter 1. Common Information Model overview ......................... 1

Chapter 2. Install the Pegasus CIM Server and base providers ......... 3
Install the OpenSSL RPM file ........................................... 3

Chapter 3. Configure the CIM Server .................................... 5
Start and stop the CIM Server .......................................... 5
Set resource limits ......................................................... 5
Secure your CIM Server ................................................. 5
Set and configure CIM Server tracing ................................ 7
Configure logging ........................................................ 8

Chapter 4. Restore and rebuild the repository ........................... 11

Chapter 5. Providers for AIX .............................................. 13
MOF files ........................................................................ 13
Providers ........................................................................ 13
Configure AIX provider tracing ......................................... 34

Appendix A. cimauth command ........................................... 35
Purpose ........................................................................... 35
Syntax ............................................................................ 35
Description ................................................................... 35
Flags ............................................................................. 35
Return values .................................................................. 35
Examples ......................................................................... 36
Related information ......................................................... 36

Appendix B. cimconfig command ........................................ 37
Purpose ........................................................................... 37
Syntax ............................................................................ 37
Description ................................................................... 37
Flags ............................................................................. 37
Exit status ...................................................................... 38
Examples ......................................................................... 38

Appendix C. cimmofo command .......................................... 39
Purpose ........................................................................... 39
Syntax ............................................................................ 39
Description ................................................................... 39
Flags ............................................................................. 40
Exit status ...................................................................... 40
Standard Error ................................................................ 40
Examples ......................................................................... 40
Related information ......................................................... 41

Appendix D. cimmofof command .......................................... 43
Purpose ........................................................................... 43

© Copyright IBM Corp. 2004, 2006
About this book

This book provides system administrators with complete information about how to perform tasks including installing, configuring, securing, and troubleshooting the Common Information Model.

Highlighting

The following highlighting conventions are used in this book:

**Bold**
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.

*Italicics*
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 5L 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 Microsoft 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 resources and populate the CIM objects described in the conceptual CIM model. These objects are then ready to be served by the CIMOM to the client management applications for managing the resources of the underlying platform. This mechanism provides an open-standard way for a management application to manage the resources of the underlying platform.

CIM on AIX 5.3 includes the following features:
• An open-source implementation of the CIMOM called Pegasus Version 2.5
• A CIM schema, version 2.9, that defines an information model for representing system management resources
• Providers that instrument a set of AIX resources and AIX diagnostics events based on CIM schema, version 2.9

For additional information about Pegasus and WBEM, see the following Web sites:
• The OpenPegasus Web site at [http://www.openpegasus.org/](http://www.openpegasus.org/)
• The DMTF Web site at [http://www.dmtf.org/standards/cim](http://www.dmtf.org/standards/cim)
• The DMTF Web site at [http://www.dmtf.org/standards/wbem](http://www.dmtf.org/standards/wbem)
Chapter 2. Install the Pegasus CIM Server and base providers

AIX 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 5.3 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
Fileset       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: Fileset 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
Fileset       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: Fileset sysmgt.pegasus.osbaseproviders not installed.
```

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 Toolbox for Linux Applications CD. It can also be downloaded from the AIX Toolbox for Linux Applications Web site at [http://www-03.ibm.com/servers/aix/products/aixos/linux/download.html](http://www-03.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.
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 \texttt{libssl.a} library to be available on the system for the CIM Server to run even if you decide not to use SSL. The \texttt{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."

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:

\begin{itemize}
  \item \texttt{cimserver} \hspace{1em} Starts the CIM Server
  \item \texttt{cimserver -s} \hspace{1em} Stops the CIM Server
\end{itemize}

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 \texttt{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.

Set resource limits

To maximize the CIM Server’s processing capacity, the root user’s \texttt{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 \texttt{smitty} at a command prompt.
2. Select \texttt{Change/Show Characteristics of a User}.
3. Type root in the \texttt{User NAME} field.
4. Scroll to the \texttt{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 \texttt{ulimit} command in the Commands Reference.

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 need 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:

```bash
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`

The cp cert.pem client.pem command creates the client.pem trust file, which is a copy of the self-signed test certificate. Its default location is the /opt/freeware/cimom/pegasus/etc directory.

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 are set in one of the following places:

- `the cimconfig command. For more information about the cimconfig command, see Appendix B, “cimconfig command,” on page 37.`
- `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 49.`
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. The components that you can specify with this parameter are:

- AsyncOpNode
- Authentication
- Authorization
- BinaryMessageHandler
- CIMExportRequestDispatcher
- CIMOMHandle
- Channel
- CimData
- Config
- ConfigurationManager
- ControlProvider
- CQL
- DiscardedData
- Dispatcher
- ExportClient
- Http
- IndDelivery
- IndHandler
- IndicationHandlerService
- IndicationService
- IndicationServiceInternal
- IPC
- L10N
- Listener
- Memory
- MessageQueueService
- MetaDispatcher
- ObjectResolution
• OsAbstraction
• ProviderAgent
• ProviderManager
• ProvManager
• Registration
• Repository
• Server
• Shutdown
• SSL
• SubscriptionService
• Thread
• UserManager
• WQL
• XmlIO
• XmlParser
• XmlReader
• XmlWriter

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:

• 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 37.

---

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. Restore and rebuild the repository

The CIM Repository is located at `/opt/freeware/cimom/pegasus/etc/repository`, and is pre-loaded with the CIM Schema, and the AIX extended classes of the CIM Schema. This repository is the active repository that can be modified with the `cimmof` and `cimmofl` commands.

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 to its original state by using either of the following options:

- **Restore the repository.** This option is recommended for all users.
  
  To restore your repository, completing the following steps:
  1. Verify that the CIM Server is not running. To stop the CIM Server, type the following command:

  ```bash
  cimserver -s
  ```

  For more information about stopping the CIM Server, see Appendix F, “cimserver command,” on page 49.
  2. Remove the active repository by typing the following command:

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

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

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

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

  This option restores the repository to the original state that was shipped with the package. Any additional classes that were 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 39 and Appendix D, “cimmofl command,” on page 43.

- **Rebuild the repository.**
  
  To rebuild your repository, complete the following steps:
  1. Verify that the CIM Server is not running. To stop the CIM Server, type the following command:

  ```bash
  cimserver -s
  ```

  For more information about stopping the CIM Server, see Appendix F, “cimserver command,” on page 49.
  2. Run the `rebuild_repository` script by typing the following command:

  ```bash
  /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 rebuilt, start the CIM Server. For more information about starting the CIM Server, see Appendix F, “cimserver command,” on page 49.

  This option restores the repository to the original state that was shipped with the package. Any additional classes that were 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 39 and Appendix D, “cimmofl command,” on page 43.
Chapter 5. Providers for AIX

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

MOF files

A .mof file is a text file that defines the class name and attributes of a managed resource. The format adheres to the CIM standard of the DMTF industry consortium. An example of a managed resource is an operating system, and the MOF file can include the following types of attributes:

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

The .mof files that define the IBM providers are located in the /usr/pegasus/provider directory.

Providers

The following provider types are included in the sysmg.t.pegasus.osbaseproviders package and are used by the Pegasus CIM Server:

Instance Providers
Provide a dynamic list of instances of a class 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 instances of different classes 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.

Indication Providers
Generates indications (events) for the defined error conditions in a specific resource (for example, the operating system).

The following CIM classes have been implemented by IBM-supplied providers.

<table>
<thead>
<tr>
<th>CIM Class</th>
<th>IBM provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMAIX_AFS</td>
<td>OSBase_AFSPublisher&quot; on page 15</td>
</tr>
<tr>
<td>IBMAIX_BlockStorageStatisticalData</td>
<td>OSBase_BlockStorageStatisticalDataProvider&quot; on page 15</td>
</tr>
<tr>
<td>IBMAIX_BootOSFromFS</td>
<td>OSBase_BootOSFromFSPublisher&quot; on page 16</td>
</tr>
<tr>
<td>IBMAIX_CDFS</td>
<td>OSBase_CDFSPublisher&quot; on page 16</td>
</tr>
<tr>
<td>IBMAIX_ComputerSystem</td>
<td>OSBase_ComputerSystemPublisher&quot; on page 17</td>
</tr>
<tr>
<td>IBMAIX_CSNetworkPort</td>
<td>OSBase_CSNetworkPortPublisher&quot; on page 16</td>
</tr>
<tr>
<td>IBMAIX_CSPProcessor</td>
<td>OSBase_CSPublisher&quot; on page 17</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2004, 2006
<table>
<thead>
<tr>
<th>CIM Class</th>
<th>IBM provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBMAIX_DFS</td>
<td>&quot;OSBase_DFSProvider&quot; on page 17</td>
</tr>
<tr>
<td>IBMAIX_EthernetPort</td>
<td>&quot;OSBase_EthernetPortProvider&quot; on page 18</td>
</tr>
<tr>
<td>IBMAIX_FCPort</td>
<td>&quot;OSBase_FCPortProvider&quot; on page 18</td>
</tr>
<tr>
<td>IBMAIX_HostedFileSystem</td>
<td>&quot;OSBase_HostedFileSystemProvider&quot; on page 19</td>
</tr>
<tr>
<td>IBMAIX_IPProtocolEndpoint</td>
<td>&quot;OSBase_IPProtocolEndpointProvider&quot; on page 19</td>
</tr>
<tr>
<td>IBMAIX_JFS</td>
<td>&quot;OSBase_JFSProvider&quot; on page 20</td>
</tr>
<tr>
<td>IBMAIX_JFS2</td>
<td>&quot;OSBase_JFS2Provider&quot; on page 20</td>
</tr>
<tr>
<td>IBMAIX_LocalLoopbackPort</td>
<td>&quot;OSBase_LoopBackPortProvider&quot; on page 21</td>
</tr>
<tr>
<td>IBMAIX_NetworkPortImplementsIPEndpoint</td>
<td>&quot;OSBase_NetworkPortImplementsEndpointProvider&quot; on page 22</td>
</tr>
<tr>
<td>IBMAIX_NFS</td>
<td>&quot;OSBase_NFSProvider&quot; on page 22</td>
</tr>
<tr>
<td>IBMAIX_OperatingSystem</td>
<td>&quot;OSBase_OperatingSystemProvider&quot; on page 22</td>
</tr>
<tr>
<td>IBMAIX_OperatingSystemStatisticalData</td>
<td>&quot;OSBase_OperatingSystemStatisticalDataProvider&quot; on page 23</td>
</tr>
<tr>
<td>IBMAIX_OSProcess</td>
<td>&quot;OSBase_OSProcessProvider&quot; on page 22</td>
</tr>
<tr>
<td>IBMAIX_Processor</td>
<td>&quot;OSBase_ProcessorProvider&quot; on page 24</td>
</tr>
<tr>
<td>IBMAIX_PROCFS</td>
<td>&quot;OSBase_PROCFSProvider&quot; on page 24</td>
</tr>
<tr>
<td>IBMAIX_RunningOS</td>
<td>&quot;OSBase_RunningOSProvider&quot; on page 25</td>
</tr>
<tr>
<td>IBMAIX_TokenRingPort</td>
<td>&quot;OSBase_TokenRingPortProvider&quot; on page 25</td>
</tr>
<tr>
<td>IBMAIX_UnixProcess</td>
<td>&quot;OSBase_UnixProcessProvider&quot; on page 26</td>
</tr>
<tr>
<td>IBMP_CSServiceEventIndication</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_Baseboard</td>
<td>&quot;IBMPSG_BaseboardProvider&quot; on page 27</td>
</tr>
<tr>
<td>IBMPSG_BIOS</td>
<td>&quot;IBMPSG_BIOSProvider&quot; on page 26</td>
</tr>
<tr>
<td>IBMPSG_CacheMemory</td>
<td>&quot;IBMPSG_CacheMemoryProvider&quot; on page 27</td>
</tr>
<tr>
<td>IBMPSG_ComputerSystem</td>
<td>&quot;IBMPSG_ComputerSystemProvider&quot; on page 28</td>
</tr>
<tr>
<td>IBMPSG_FanEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_FRU</td>
<td>&quot;IBMPSG_FRUProvider&quot; on page 28</td>
</tr>
<tr>
<td>IBMPSG_LogicalDisk</td>
<td>&quot;IBMPSG_LogicalDiskProvider&quot; on page 29</td>
</tr>
<tr>
<td>IBMPSG_MemoryPFEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_NetworkAdapterFailedEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_OperatingSystem</td>
<td>&quot;IBMPSG_OperatingSystemProvider&quot; on page 30</td>
</tr>
<tr>
<td>IBMPSG_PowerSupplyEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_Processor</td>
<td>&quot;IBMPSG_ProcessorProvider&quot; on page 32</td>
</tr>
<tr>
<td>IBMPSG_ProcessorPFEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_Slot</td>
<td>&quot;IBMPSG_SlotProvider&quot; on page 32</td>
</tr>
<tr>
<td>IBMPSG_SystemEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_TemperatureEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
<tr>
<td>IBMPSG_VoltageEvent</td>
<td>&quot;IBMPSG_IndicationProvider&quot; on page 33</td>
</tr>
</tbody>
</table>
**OSBase_AFSProvider**

This provider is an instance provider for the AIX_AFS class. It conforms to the CMPI interface to provide information about remote AFS file systems.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**OSBase_BlockStorageStatisticalDataProvider**

This provider is a statistical provider for the IBMAIX_BlockStorageStatisticalData class. It provides statistical data for the block storage system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalIOs</td>
<td>uint64</td>
</tr>
<tr>
<td>KBytesTransferred</td>
<td>uint64</td>
</tr>
<tr>
<td>IOTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadIOs</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadHitIOs</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadIOTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadHitIOTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>KBytesRead</td>
<td>uint64</td>
</tr>
<tr>
<td>WriteIOs</td>
<td>uint64</td>
</tr>
<tr>
<td>WriteHitIOs</td>
<td>uint64</td>
</tr>
<tr>
<td>WriteIOTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>WriteHitIOTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>IdleTimeCounter</td>
<td>uint64</td>
</tr>
<tr>
<td>MaintOp</td>
<td>uint64</td>
</tr>
<tr>
<td>MaintTimeCounter</td>
<td>uint64</td>
</tr>
</tbody>
</table>
For more information about the properties descriptions, see the IBMAIX_BlockStorageStatisticalData class in the `/usr/pegasus/provider/mof/IBMAIX_OSBase.mof` file.

**OSBase_BootOSFromFSProvider**

This provider is an instance provider for the IBMAIX_BootOSFromFS class. It conforms to the CMPI interface to provide 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 for the IBMAIX_CDFS class. It conforms to the CMPI interface to provide information about local CDFS file systems.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 instance provider for the IBMAIX_CSNetworkPort class. It conforms to the CMPI interface to provide 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 instance provider for the IBMAIX_CSProcessor class. It conforms to the CMPI interface to provide 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 for the IBMAIX_ComputerSystem class. It conforms to the CMPI interface to provide computer system properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>NameFormat</td>
<td>string</td>
</tr>
<tr>
<td>Dedicated[]</td>
<td>uint16A</td>
</tr>
<tr>
<td>OtherDedicatedDescriptions</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_DFS class. It conforms to the CMPI interface to provide information about remote DFS file systems.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_EthernetPort class. It conforms to the CMPI interface to provide the ethernet port properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>Element Name</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemCreationClassname</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>MaxSpeed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>Speed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>OtherNetworkPortType</td>
<td>string</td>
</tr>
<tr>
<td>LinkTechnology</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherLinkTechnology</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_FCPort class. It conforms to the CMPI interface to provide fibre channel port properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
</tbody>
</table>

**AIX 5L Version 5.3 Common Information Model Guide**
### Property Type

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Name</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>MaxSpeed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>Speed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>OtherNetworkPortType</td>
<td>string</td>
</tr>
<tr>
<td>LinkTechnology</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherLinkTechnology</td>
<td>string</td>
</tr>
</tbody>
</table>

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 instance provider for the IBMAIX_HostedFileSystem class. It conforms to the CMPI interface to provide 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 for the IBMAIX_IPProtocolEndpoint class. It conforms to the CMPI interface to provide IP protocol end point properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>NameFormat</td>
<td>string</td>
</tr>
<tr>
<td>Protocol Type</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherTypeDescription</td>
<td>string</td>
</tr>
<tr>
<td>IPv4Address</td>
<td>string</td>
</tr>
<tr>
<td>IPv6Address</td>
<td>string</td>
</tr>
<tr>
<td>SubnetMask</td>
<td>string</td>
</tr>
<tr>
<td>PrefixLength</td>
<td>uint8</td>
</tr>
<tr>
<td>IPVersionSupport</td>
<td>uint16</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_JFS2 class. It conforms to the CMPI interface to provide information about local enhanced journaled file system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_JFS class. It conforms to the CMPI interface to provide information about local journaled file system.
The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_LoopBackPort class. It conforms to the CMPI interface to provide loopback port properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>LinkTechnology</td>
<td>string</td>
</tr>
<tr>
<td>OtherLinkTechnology</td>
<td>uint16</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_NFS class. It conforms to the CMPI interface to provide information about remote NFS file systems.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 instance provider for the IBMAIX_NetworkPortImplementsEndpoint class. It conforms to the CMPI interface to provide 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, token-ring, 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 instance provider for the IBMAIX_OSProcess class. It conforms to the CMPI interface to provide 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 for the IBMAIX_OperatingSystem class. It conforms to the CMPI interface to provide AIX operating system properties.
The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>OSType</td>
<td>uint16</td>
</tr>
<tr>
<td>Version</td>
<td>string</td>
</tr>
<tr>
<td>LastBootUpTime</td>
<td>datetime</td>
</tr>
<tr>
<td>LocalDateTime</td>
<td>datetime</td>
</tr>
<tr>
<td>CurrentTimeZone</td>
<td>sint16</td>
</tr>
<tr>
<td>NumberOfProcesses</td>
<td>uint32</td>
</tr>
<tr>
<td>MaxNumberOfProcesses</td>
<td>uint32</td>
</tr>
<tr>
<td>TotalVirtualMemorySize</td>
<td>uint64</td>
</tr>
<tr>
<td>FreeVirtualMemory</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>FreePhysicalMemory</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>TotalVisibleMemorySize</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>SizedStoredInPagingFiles</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>FreeSpaceInPagingFiles</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>MaxProcessorsMemorySize</td>
<td>uint64 (in KBytes)</td>
</tr>
<tr>
<td>Distributed</td>
<td>Boolean</td>
</tr>
<tr>
<td>CodeSet</td>
<td>string</td>
</tr>
<tr>
<td>LanguageEdition</td>
<td>string</td>
</tr>
<tr>
<td>DefaultPageSize</td>
<td>uint32</td>
</tr>
</tbody>
</table>

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

**OSBase_OperatingSystemStatisticalDataProvider**

This provider is a statistical provider for the IBMAIX_OperatingSystemStatisticalData class and provides statistical data for the operating system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUUserTime</td>
<td>uint64</td>
</tr>
<tr>
<td>CPUSystemTime</td>
<td>uint64</td>
</tr>
<tr>
<td>CPUWaitTime</td>
<td>uint64</td>
</tr>
<tr>
<td>CPUIdleTime</td>
<td>uint64</td>
</tr>
<tr>
<td>RunQueueLength</td>
<td>uint64</td>
</tr>
<tr>
<td>BlockQueueLength</td>
<td>uint64</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td>PagesPagedIn</td>
<td>uint64</td>
</tr>
<tr>
<td>PagesPagedOut</td>
<td>uint64</td>
</tr>
</tbody>
</table>

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

**OSBase_PROCFSProvider**

This provider is an instance provider for the IBMAIX_PROCFS class. It conforms to the CMPI interface to provide information about local PROCFS file system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Root</td>
<td>string</td>
</tr>
<tr>
<td>FilesystemSize</td>
<td>uint64</td>
</tr>
<tr>
<td>AvailableSpace</td>
<td>uint64</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Boolean</td>
</tr>
<tr>
<td>FilesystemType</td>
<td>string</td>
</tr>
</tbody>
</table>

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 for the IBMAIX_Processor class. It conforms to the CMPI interface to provide individual process properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>Role</td>
<td>string</td>
</tr>
<tr>
<td>Family</td>
<td>uint16</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>OtherFamilyDescription</td>
<td>string</td>
</tr>
<tr>
<td>MaxClockSpeed</td>
<td>uint32 (Mhz)</td>
</tr>
<tr>
<td>CurrentClockSpeed</td>
<td>uint32 (Mhz)</td>
</tr>
<tr>
<td>LoadPercentage</td>
<td>uint16</td>
</tr>
<tr>
<td>Stepping</td>
<td>string</td>
</tr>
<tr>
<td>CPU Status</td>
<td>uint16</td>
</tr>
</tbody>
</table>

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 instance provider for the IBMAIX_RunningOS class. It conforms to the CMPI interface
to provide 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 for the IBMAIX_TokenRingPort class. It conforms to the CMPI
interface to provide token-ring port properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>Element Name</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>MaxSpeed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>Speed</td>
<td>uint64 (bps)</td>
</tr>
<tr>
<td>OtherNetworkPortType</td>
<td>string</td>
</tr>
<tr>
<td>LinkTechnology</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherLinkTechnology</td>
<td>string</td>
</tr>
</tbody>
</table>
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 for the IBMAIX_UnixProcess class. It conforms to the CMPI interface to provide individual process properties.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>OSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>OSName</td>
<td>string</td>
</tr>
<tr>
<td>Handle</td>
<td>string</td>
</tr>
<tr>
<td>CreationDate</td>
<td>datetime</td>
</tr>
<tr>
<td>KernelModeTime</td>
<td>uint64</td>
</tr>
<tr>
<td>UserModeTime</td>
<td>uint64</td>
</tr>
<tr>
<td>ParentProcessID</td>
<td>string</td>
</tr>
<tr>
<td>RealUserID</td>
<td>uint64</td>
</tr>
<tr>
<td>ProcessGroupID</td>
<td>uint64</td>
</tr>
<tr>
<td>ProcessTTY</td>
<td>string</td>
</tr>
<tr>
<td>ModulePath</td>
<td>string</td>
</tr>
<tr>
<td>Parameters[]</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_BIOSProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about BIOS on the system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMBIOSBIOSVersion</td>
<td>string</td>
</tr>
<tr>
<td>Version</td>
<td>string</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>string</td>
</tr>
<tr>
<td>PrimaryBIOS</td>
<td>boolean</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Version</td>
<td>string</td>
</tr>
<tr>
<td>TargetOperatingSystem</td>
<td>uint16</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SMBIOSPresent</td>
<td>boolean</td>
</tr>
</tbody>
</table>

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

**IBMPSG_BaseboardProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the baseboard. This provider is used by the IBM® Director agent.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoweredOn</td>
<td>boolean</td>
</tr>
<tr>
<td>Tag</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>string</td>
</tr>
<tr>
<td>Model</td>
<td>string</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_CacheMemoryProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about cache memory on the system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>InstalledSize</td>
<td>uint32</td>
</tr>
</tbody>
</table>

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

**IBMPSG_ComputerSystemProvider**

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

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>string</td>
</tr>
<tr>
<td>NameFormat</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>PrimaryOwnerName</td>
<td>string</td>
</tr>
<tr>
<td>PrimaryOwnerContact</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>LPARID</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_FRUPProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the FRUs on the system.
The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRUNumber</td>
<td>string</td>
</tr>
<tr>
<td>IdentifyingNumber</td>
<td>string</td>
</tr>
<tr>
<td>Vendor</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_LogicalDiskProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the logical disks on the system. This provider is used by the IBM Director agent.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataOrganization</td>
<td>uint16</td>
</tr>
<tr>
<td>Purpose</td>
<td>string</td>
</tr>
<tr>
<td>Access</td>
<td>uint16</td>
</tr>
<tr>
<td>BlockSize</td>
<td>uint64</td>
</tr>
<tr>
<td>NumberOfBlocks</td>
<td>uint64</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>Availability</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

For more information about the properties descriptions, see the IBMPSG_LogicalDisk class in the /usr/pegasus/provider/mof/IBMPSG_LogicalSystem.mof file.
**IBMPSG_OperatingSystemProvider**

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

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeSet</td>
<td>string</td>
</tr>
<tr>
<td>LanguageEdition</td>
<td>string</td>
</tr>
<tr>
<td>CSCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>CSName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>OSType</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherTypeDescription</td>
<td>string</td>
</tr>
<tr>
<td>Version</td>
<td>string</td>
</tr>
<tr>
<td>LastBootUpTime</td>
<td>datetime</td>
</tr>
<tr>
<td>LocalDateTime</td>
<td>datetime</td>
</tr>
<tr>
<td>CurrentTimeZone</td>
<td>sint16</td>
</tr>
<tr>
<td>NumberOfLicensedUsers</td>
<td>uint32</td>
</tr>
<tr>
<td>NumberOfUsers</td>
<td>uint32</td>
</tr>
<tr>
<td>NumberOfProcesses</td>
<td>uint32</td>
</tr>
<tr>
<td>TotalVirtualMemorySize</td>
<td>uint64</td>
</tr>
<tr>
<td>FreeVirtualMemory</td>
<td>uint64</td>
</tr>
<tr>
<td>FreePhysicalMemory</td>
<td>uint64</td>
</tr>
<tr>
<td>TotalVisibleMemorySize</td>
<td>uint64</td>
</tr>
<tr>
<td>SizeStoredInPagingFiles</td>
<td>uint64</td>
</tr>
<tr>
<td>FreeSpaceInPagingFiles</td>
<td>uint64</td>
</tr>
<tr>
<td>MaxProcessMemorySize</td>
<td>uint64</td>
</tr>
<tr>
<td>Distributed</td>
<td>boolean</td>
</tr>
<tr>
<td>MaxProcessesPerUser</td>
<td>uint32</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
</tbody>
</table>

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

**IBMPSG_PhysicalDiskProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the physical disks on the system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxMediaSize</td>
<td>uint64</td>
</tr>
<tr>
<td>DefaultBlockSize</td>
<td>uint64</td>
</tr>
</tbody>
</table>

AIX 5L Version 5.3 Common Information Model Guide
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnitsDescription</td>
<td>string</td>
</tr>
<tr>
<td>MaxUnitsBeforeCleaning</td>
<td>uint64</td>
</tr>
<tr>
<td>UnitsUsed</td>
<td>uint64</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_PhysicalMemoryProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the physical memory on the system. This provider is used by the IBM Director agent.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>uint64</td>
</tr>
<tr>
<td>DeviceLocator</td>
<td>string</td>
</tr>
<tr>
<td>PartNumber</td>
<td>string</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>string</td>
</tr>
<tr>
<td>Capacity</td>
<td>uint64</td>
</tr>
<tr>
<td>Tag</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>string</td>
</tr>
<tr>
<td>Version</td>
<td>string</td>
</tr>
<tr>
<td>PartNumber</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>
For more information about the properties descriptions, see the IBMPSG_ComputerSystem class in the /usr/pegasus/provider/mof/IBMPSG_PhysicalMemory.mof file.

**IBMPSG_ProcessorProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the processors on the system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>string</td>
</tr>
<tr>
<td>Family</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherFamilyDescription</td>
<td>string</td>
</tr>
<tr>
<td>MaxClockSpeed</td>
<td>uint32</td>
</tr>
<tr>
<td>CurrentClockSpeed</td>
<td>uint32</td>
</tr>
<tr>
<td>LoadPercentage</td>
<td>uint16</td>
</tr>
<tr>
<td>Stepping</td>
<td>string</td>
</tr>
<tr>
<td>CPUStatus</td>
<td>uint16</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>DeviceID</td>
<td>string</td>
</tr>
<tr>
<td>EnabledState</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherEnabledState</td>
<td>string</td>
</tr>
<tr>
<td>RequestedState</td>
<td>uint16</td>
</tr>
<tr>
<td>EnabledDefault</td>
<td>uint16</td>
</tr>
<tr>
<td>Name</td>
<td>string</td>
</tr>
<tr>
<td>Status</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_SlotProvider**

This provider is an instance provider that writes to the CMPI interface and provides information about the slots on the system.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectorType[]</td>
<td>uint16</td>
</tr>
<tr>
<td>SupportsHotPlug</td>
<td>boolean</td>
</tr>
<tr>
<td>OtherTypeDescription</td>
<td>string</td>
</tr>
<tr>
<td>Tag</td>
<td>string</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>CreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
<tr>
<td>Caption</td>
<td>string</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>ElementName</td>
<td>string</td>
</tr>
</tbody>
</table>

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

**IBMPSG_IndicationProvider**

This provider generates CIM indications of the class types and properties in the following list:

- **IBMPSG_FanEvent** indication is generated when a fan device error is detected by an AIX® diagnostic event.
- **IBMPSG_VoltageEvent** indication is generated when a voltage alert is detected by an AIX diagnostic event (JS20 only).
- **IBMPSG_TemperatureEvent** indication is generated when a temperature alert is detected by an AIX diagnostic event (JS20 only).
- **IBMPSG_NetworkAdapterFailedEvent** indication is generated when a network adapter failed error is detected by an AIX diagnostic event.
- **IBMPSG_PowerSupplyEvent** indication is generated when a power supply error is detected by an AIX diagnostic event.
- **IBMPSG_ProcessorPFEvent** indication is generated when a processor error is detected by an AIX diagnostic event.
- **IBMPSG_MemoryPFEvent** indication is generated when a memory error is detected by an AIX diagnostic event.
- **IBM_SystemEvent** indication is generated for all other errors detected by an AIX diagnostic event. For this indication, the EventID property contains the resource name that caused the error.
- **IBMP_CSServiceEventIndication** is generated when the errors are serviced by a customer engineer. For this indication, the EventID property contains the resource name that was serviced.

The following provider properties are implemented:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>string</td>
</tr>
<tr>
<td>SystemCreationClassName</td>
<td>string</td>
</tr>
<tr>
<td>SystemName</td>
<td>string</td>
</tr>
<tr>
<td>IndicationIdentifier</td>
<td>string</td>
</tr>
<tr>
<td>IndicationTime</td>
<td>dateTime</td>
</tr>
<tr>
<td>AlertingManagedElement</td>
<td>string</td>
</tr>
<tr>
<td>EventID</td>
<td>string</td>
</tr>
<tr>
<td>AlertType</td>
<td>uint16</td>
</tr>
<tr>
<td>PerceivedSeverity</td>
<td>uint16</td>
</tr>
<tr>
<td>UniqueSystemID</td>
<td>string</td>
</tr>
<tr>
<td>Trending</td>
<td>uint16</td>
</tr>
<tr>
<td>OtherAlertType</td>
<td>string</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ProbableCause</td>
<td>uint16</td>
</tr>
<tr>
<td>ProbableCauseDescription</td>
<td>string</td>
</tr>
<tr>
<td>RecommendedActions</td>
<td>string</td>
</tr>
</tbody>
</table>

For more information about the properties descriptions, see the `/usr/pegasus/provider/mof/IBMPSG_AlertIndication.mof` file.

### 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=path`, where `path` 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:

```bash
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
-c
-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.

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.
- `-I` Displays the authorizations of all CIM users.
- `-m` Modifies the authorizations for a user on a namespace.
- `-n namespace`
  Specifies the namespace for which you are adding authorizations to users.
  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.
- `-r`
  Removes the authorizations for a user on a namespace.
- `-R`
  Grants read authorization.
- `-u username`
  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, unsets, 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

- **-c**
  Indicates that the action taken applies to the current configuration property value. This flag must be used in conjunction with one of the -g, -s, -u, or -l flags.

- **-d**
  Indicates that the action taken applies to the default configuration property value. This flag must be used in conjunction with the -g flag.

- **-g name**
  Displays the current, planned, or default value of the specified configuration property depending on whether the -c, -p, or -d flag is specified. If no other flag is specified, the current value of the specified configuration property is displayed.

- **-l**
  Lists the names of all the configuration properties when used without any other flag. When this flag is paired with the -c flag, a paired listing of all the current configuration properties and their values is displayed. When this flag is paired with the -p flag, a paired listing of all the planned configuration properties and their values is displayed.

- **-p**
  Indicates that the action taken applies to the planned configuration property value. This flag must be used in conjunction with one of the -g, -s, -u, or -l flags.

- **-s name=value**
  Sets the current or planned value of the specified configuration property to the specified 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 set to the specified value. The name parameter is the name of the configuration property, and value is the new value for the configuration property. An error message is returned if the specified property cannot be set dynamically.

© Copyright IBM Corp. 2004, 2006
 Resets the current or planned value of the specified configuration property to the default value depending on whether the \texttt{-c} or \texttt{-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**

<table>
<thead>
<tr>
<th>Exit Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully.</td>
</tr>
<tr>
<td>1</td>
<td>The command did not complete successfully, and an explanatory message is written to stderr.</td>
</tr>
</tbody>
</table>

**Examples**

1. To view the current value for the \texttt{port} configuration property, type one of the following commands:
   
   ```bash
   cimconfig -g port
   cimconfig -g port -c
   ```

2. To view the planned value of the \texttt{traceLevel} configuration property, type the following:
   
   ```bash
   cimconfig -g traceLevel -p
   ```

3. To change the current value of the \texttt{traceLevel} configuration property to the value 2, type one of the following commands:
   
   ```bash
   cimconfig -s traceLevel=2
   cimconfig -s traceLevel=2 -c
   ```

4. To change the planned value of the \texttt{traceLevel} property to a value of 3, type the following:
   
   ```bash
   cimconfig -s traceLevel=3 -p
   ```

5. To reset the current value of the \texttt{traceLevel} property to the default value, type one of the following commands:
   
   ```bash
   cimconfig -u traceLevel
   cimconfig -u traceLevel -c
   ```

6. To reset the planned value of the \texttt{traceLevel} property to the default value, type the following:
   
   ```bash
   cimconfig -u traceLevel -p
   ```

7. To list all of the current configuration properties and their values, type the following:
   
   ```bash
   cimconfig -l -c
   ```

8. To list all of the planned configuration properties and their values, type the following:
   
   ```bash
   cimconfig -l -p
   ```
Appendix C. cimmof command

Purpose
Compiles MOF files into the CIM repository through the CIM Server.

Syntax

```
cimmof [-h]
cimmof [-w] [-I 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 `-I` 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

-h Displays the command usage and the version number of the MOF compiler.
-I path 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.
-n namespace 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 root/PG_InterOp namespace.
-w 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 cimmof command returns warning messages. Use this flag to suppress those warning messages.

Exit status

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

Standard 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 cimmof 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 a MOF file with one or more superclasses that are not in the CIM Repository.

Examples

1. To compile an MOF file named processinfo.mof into the default namespace in the CIM Repository, type the following:
cimmof processinfo.mof
2. To compile the MOF file name processinfo.mof into the root/application namespace, type the following:
cimmof -n root/application processinfo.mof
3. 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 **cimmosf** command and display the version of the MOF Compiler, type the following:

   ```cimmosf -h```

---

**Related information**

The **cimserver** command.
Appendix D. cimmofl command

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

Syntax
```
cimmofl [-h]
cimmofl [-w] [-I path] [-n namespace] file...file
```

Description
The cimmofl 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 cimmofl 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 cimmofl 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 cimmofl 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 request 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.

© Copyright IBM Corp. 2004, 2006
Flags

- **-h**
  Displays the command usage and the version number of the MOF compiler.

- **-I path**
  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.

- **-n namespace**
  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 `root/PG_InterOp` namespace.

- **-w**
  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 `cimmofl` command returns warning messages. Use this flag to suppress those warning messages.

Exit status

0
The command completed successfully.

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

Standard Error

Error

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

Description

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

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

Examples

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

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

3. 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
   ```

4. To list the arguments to the `cimmofl` command and display the version of the MOF Compiler, type the following:
cimomi -h

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
- `-d` 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.

- `-e` 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.

© Copyright IBM Corp. 2004, 2006
Displays all the registered provider modules when it is not used with the -m module option.

Displays all the providers in the specified provider module when used with the -m module option.

Specifies the provider module for the operation.

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 cimmoF command.

Displays the status of provider modules.

The command completed successfully.

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

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 module, 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

The cimmof and cimserver commands.
Appendix F. cimserver command

Purpose
Starts and stops the CIM Server.

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

Description
The `cimserver` command allows you to start and stop the CIM Server.

Flags
- `-h` Prints the usage statement for this command.
- `-s` Stops the CIM Server.
- `-v` Displays the CIM Server version number.

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

Exit status
- `0` The command completed successfully.
- `>0` The command did not complete successfully, and an explanatory message is written to stderr.

Examples
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 37.

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 35.

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

Flags

- a
  Adds a CIM user.
- l
  Lists all current CIM users.
- m
  Modifies the password for the user specified in the -u username option.
  - n newpassword
    Identifies the new password for the user specified in the -u username option.
- r
  Removes a CIM user. Any authorizations for the specified user will be deleted.
- u username
  Identifies the user you want to add, remove, or for whom you want to modify the password.
- w password
  Identifies the password for the user specified in the -u username 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 106-0032, 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
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- UNIX® is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be the trademarks or service marks of others.
- Linux
- Microsoft
Index

A
About this book v
AIX provider tracing 34
AIX providers 13

C
Case-sensitivity in AIX v
CIM classes 13
CIM repository 9
  restoring and rebuilding 9
CIM Server
  configuring 4
  enabling with SSL 5
  installing 3
  securing 5
  setting and configuring 7
  setting resource limits 4
  starting and stopping 4
  tracing 7
  cimauth command 35
  cimconfig command 37
  cimmof command 39
  cimmofl command 43
  CIMOM i
  cimprovider command 47
  cimserver command 49
  cimuser command 51
  classes 13
  CMPI 13
  commands
    cimauth 35
    cimconfig 37
    cimmof 39
    cimmofl 43
    cimprovider 47
    cimserver 49
    cimuser 51
    sed 5
  Common Information Model
    features i
    Object Manager i
    overview i
  configuration files 4
  configuring
    CIM Server 4
    http port 4
    https port 4
    logging 8
    SSL mode 4

D
DFS 17
Distributed Management Task Force i
DMTF i

H
Highlighting conventions in this book v

I
install
  filesets 3
  installp command 3
  procedure 3
Installing the OpenSSL RPM file 3
Installing the Pegasus CIM Server and base providers 3
ISO 9000 v

L
logging 8
logLevel parameter 8

M
management applications i
management systems i
MOF files 13

O
OpenSSL RPM file
  installing 3
  OSBase_AFSProvider 15
  OSBase_BlockStorageStatisticalDataProvider 15
  OSBase_BootOSFromFSProvider 16
  OSBase_CDFSProvider 16
  OSBase_ComputerSystemProvider 17
  OSBase_CSNetworkPortProvider 16
  OSBase_CSProcessorProvider 16
  OSBase_DFSProvider 17
  OSBase_EthernetPortProvider 18
  OSBase_FCPortProvider 18
  OSBase_HostedFileSystemProvider 19
  OSBase_IPProtocolEndpointProvider 19
  OSBase_JFS2Provider 20

P
Pegasus i
  installing 3
  provider tracing 34
  configuring 34
  providers 13

R
repository 9
Resource limits
  setting 4
S
schema i
secure server 5
security 5
sed command 5
SSL 5
certificates 5
enable 5
keys 5
mode 5
public key 5
RSA key 5
SSL certificates
creating 5

T
tracing 7

W
WBEM i
Web-based Enterprise Management i
Readers’ Comments — We’d Like to Hear from You

AIX 5L Version 5.3
Common Information Model Guide

Publication No. SC23-4942-02

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

<table>
<thead>
<tr>
<th>Overall satisfaction</th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Very Satisfied</th>
<th>Satisfied</th>
<th>Neutral</th>
<th>Dissatisfied</th>
<th>Very Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicable to your tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Name

Company or Organization

Phone No.
Readers' Comments — We'd Like to Hear from You

SC23-4942-02

IBM Corporation
Information Development
Department 04XA-905-6C006
11501 Burnet Road
Austin, TX  78758-3493

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 04XA-905-6C006
11501 Burnet Road
Austin, TX  78758-3493

SC23-4942-02