System i

Systems management
Common Information Model

Version 6 Release 1

System i

Systems management
Common Information Model

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in "Notices," on page 105.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

Common Information Model

The Common Information Model (CIM) is a standard that is developed by Distributed Management Task Force (DMTF). DMTF is a consortium of major hardware and software vendors (including IBM) that is a part of the Web-Based Enterprise Management (WBEM) initiative.

WBEM includes a set of standards and technologies that provide management solutions for a distributed network environment. Interoperability is a major focus of WBEM. WBEM technologies can help you develop a single set of management applications for a diverse set of resources.

CIM is a major component of the WBEM initiative that provides a model for describing and accessing data across an enterprise. CIM comprises both a specification and a schema. The specification defines the details for integration with other management models, whereas the schema provides the actual model descriptions.

On i5/OS® V6R1, CIM is a feature of the IBM® Universal Manageability Enablement for i5/OS licensed program (5722-UME). It includes the following functions:

- Providers instrumentation for server resources on the system. The providers, which are based on a subset of the standardized CIM classes, gather data on a system.
- Common Information Model Object Manager (CIMOM), a central component of the WBEM server that is responsible for the communication between clients and providers. CIMOM also provides several management functions, including security, and a set of commands that provide configuration and management functions to administrators.
- A schema that defines an information model for representing systems management functions.
- An implementation of the standardized formats for communication between clients and CIMOM, called CIM in XML, V2.1 and CIM operations over HTTP, V1.1. For more information about these standards, see the WBEM Web site.

CIM on i5/OS V6R1 is also supported by i5/OS V5R4.

Related information

Web-Based Enterprise Management (WBEM)

Common Information Model: Introduction to CIM

Common Information Model (CIM) Standards

The Open Group: OpenPegasus

What's new for V6R1

Read about new or significantly changed information for the Common Information Model topic collection.

IBM Universal Manageability Enablement for i5/OS licensed program enhancements

CIM has moved out of the operating system to be ported with the IBM Universal Manageability Enablement for i5/OS licensed program (5722-UME). The CIM server runs in i5/OS Portable Application Solutions Environment (i5/OS PASE).

- The following CIM functions are supported:
  - The following Common Information Model Object Manager (CIMOM) functions are supported:

    - Secure Sockets Layer (SSL)
    - Common Manageability Programming Interface (CMPI)
    - Out-of-process provider
    - Service Location Protocol (SLP)
    - Distributed Management Task Force (DMTF) schema 2.14
    - Internet Protocol version 6 (IPv6)
    - Server Profile support (OpenPegasus start support Server Profile from 2.6.0)
  - The following CIMOM functions were removed:
    - Kerberos support
    - Obsolete configuration properties
      - tempLocalAuthDir
      - enableHttpLocalconnection
      - kerberosServiceName
  - The following CIMOM functions were changed to fixed properties:
    - home
    - daemon
    - slp
    - repositoryDir
  - The -q option is no longer needed for the cimconfig and cimprovider commands in PASE. Some new properties were added to the basic startup properties and advanced startup properties for CIMOM. The cimmof and ssltrustmgr commands are available in this release.
    - "Basic startup properties for CIMOM" on page 7
    - "Advanced startup properties for CIMOM" on page 9
    - "cimmof usage information" on page 27
    - "cimconfig usage information" on page 25
    - "cimprovider usage information" on page 29
    - "ssltrustmgr usage information" on page 30
  - The following properties were changed to fixed properties, and you cannot change their values:
    - daemon
    - repositoryDir
- The following providers are supported:
  - Storage Management Initiative Specification (SMI-S) host bus adapter (HBA) and host discovered resources (HDR) CIM providers
  - "Providers that are inherited from the operating system" on page 35
  - "Hardware inventory and network management providers" on page 48

What's new as of 30 April 2008

You can use the IBM Universal Manageability Enablement for i5/OS licensed program (5722-UME) on i5/OS client partitions, with limitations. Physical hardware information is not accessible to i5/OS partitions that are managed by Integrated Virtualization Manager. See "Considerations for providers on i5/OS client partitions" on page 34 for more information.

You can use IBM Systems Director Navigator for i5/OS to start and stop CIMOM. See "Starting and stopping CIMOM" on page 16 for more information.

CIMOM has a new configuration property, idleSessionTimeout. See "Advanced startup properties for CIMOM" on page 9 for more information.

How to see what’s new or changed

To help you see where technical changes have been made, this information uses:

- The » image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

To find other information about what’s new or changed this release, see the Memo to users.

PDF file for Common Information Model

You can view and print a PDF file of this information.

To view or download the PDF version of the Common Information Model topic, select Common Information Model (about 1 400 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe® Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Related reference

“Related information for Common Information Model” on page 102
Web sites and other information center topic collections contain information that relates to the Common Information Model topic collection. You can view or print any of the PDF files.

CIM upgrade considerations

If you upgrade the i5/OS operating system from V5R4 to V6R1, the IBM Universal Manageability Enablement for i5/OS licensed program is installed by default. The CIM function that is supplied with i5/OS V5R4 is disabled.

If you use the system-based CIM function in V5R4, you must start the CIM server before you configure CIMOM after the upgrade operation. The CIM server starts a repository and configuration migration process. During the migration, symbolic links are created; therefore, you cannot use the cimconfig, cimmof, and cimprovider commands.

Before the migration is completed, the CIM server is not available to process CIM requests; do not use any client programs.

The migration might take some time, depending on the size of the repository, processor speed, and system utilization.

If you are already using the CIM function of the IBM Universal Manageability Enablement for i5/OS licensed program on your i5/OS V5R4 operating system, you can continue with CIMOM configuration directly after you upgrade your system to i5/OS V6R1.

Repository migration

The IBM Universal Manageability Enablement for i5/OS licensed program provides a repository that includes a set of files that contain the CIM class definitions, instances of classes, and provider registration instances. The repository is stored in the UserData directory (/QOpenSys/QIBM/UserData/UME/Pegasus).

The CIM repository is migrated from Distributed Management Task Force (DMTF) CIM schema V2.7 or V2.9 to DMTF CIM schema V2.14. During the migration, the CIM server is not available to process CIM requests. If you stop the server job during the migration, loss of data might occur.

The following objects are not migrated:

- CIM provider register information
- Static instances of metric definition in the repository (Providers dynamically collect information and implement the same functions as these metric instances.)

The CIM server log (by default, located at /QOpenSys/QIBM/UserData/UME/Pegasus/logs) might have the following messages related to the repository migration:

Message PGS10100

This message is written into the CIM server log when the migration begins.

```
PGS10100: The CIM server is starting to restore the repository and then migrate the repository from
an earlier version. This will take several minutes, during which the server will not be available.
Stopping the server job might result in the loss of data.
```

Message PGS10101

This message is written into the CIM server log when the migration is completed without errors.

```
PGS10101: The Common Information Model (CIM) repository has been migrated successfully.
```

Configuration migration

The CIM server migrates the configuration file of the operating system CIM server. The configuration properties are migrated and their values are not changed. However, the following configuration properties are not migrated:

- logdir, home, daemon, slp, repositoryDir, tempLocalAuthDir, and kerberosServiceName.
- enableHttpsConnection, sslClientVerificationMode, and httpAuthType if the value of property httpAuthType in the operating system CIM server is Kerberos. Property enableHttpConnection is set to false after the migration.
- enableHttpLocalConnection.

The migrated configuration properties are not checked for validity. If the configuration properties of the operating system CIM server are not set to function properly, this might prevent the new CIM server from starting and functioning successfully.

Configuring CIMOM

When you install the i5/OS operating system, the IBM Universal Manageability Enablement for i5/OS licensed program is installed on the system by default. Before you use the CIM function of this licensed program, you need to configure the Common Information Model Object Manager (CIMOM).

About this task

Note: If you upgrade the i5/OS operating system from V5R4, the CIM function that is supplied with i5/OS V5R4 is disabled. If you use the system-based CIM function in V5R4, you must start the CIM server before you configure CIMOM after the upgrade operation. The CIM server starts a repository and configuration migration process. Before the migration is completed, the CIM server is not available to process CIM requests. Do not use any client programs during the migration. See "CIM upgrade considerations" on page 3 for details.

To configure CIMOM, follow these steps:

1. Ensure that the operating system has the required installation requirements.
2. Set the configuration properties by using the cimconfig -s -p command.
3. Grant users the authorizations required to work with CIMOM. In the operating system, Application Administration controls operations that change the local CIM schema, and object authorities control operations that change the system objects.
4. Start CIMOM.

Related concepts

"CIMOM command-line utilities" on page 25
You can use a set of command-line utilities to control or change the CIMOM environment. These command-line utilities include cimmof, cimconfig, cimprovider, and ssltrustmgr.

"i5/OS metrics classes" on page 95
The topic describes metric classes and the user authorization.

"cimconfig usage information" on page 25
You can configure the startup properties for CIMOM with the cimconfig command.

Ensuring that the operating system has the required installation requirements

CIM requires specific installation options and product on the i5/OS operating system.

You must have the following programs installed on the system:

- i5/OS Portable Application Solutions Environment (i5/OS PASE) (5761-SS1, option 33)
- OpenSSH, OpenSSL, zlib functions, IBM Portable Utilities for i5/OS (5733-SC1, option 1)
- IBM Universal Manageability Enablement for i5/OS (5722-UME)

Note: In V5R4, you need to install all the required fixes. Refer to Infor APAR (5722-UME CONSIDERATIONS).

Setting the configuration properties

Before starting the CIM server, you need to set several configuration properties using the cimconfig -s -p command.

Enabling the CIM server with Secure Sockets Layer

To enable the CIM server to run in Secure Sockets Layer (SSL) mode, a private key and a certificate are required. The administrator can create the private key and certificate by signing it with a certificate authority (CA).

The CIM server checks for its private key and certificate during startup. If either of the files does not exist, the server creates its private key and a self-signed, 365-day certificate. These files are created in the location that is defined by the value of the sslCertificateFilePath and sslKeyFilePath properties.

The server creates its certificate with the following fields for the distinguished name:

- Country Name: US
- State or Province Name: Minnesota
- Locality: Rochester
- Organization Name: IBM
- Organizational Unit: System i®
- Common Name: *hostname of the system*
- Email Address:

Note: The Common Name is replaced by the hostname of this system, and the Email Address is left blank.

Creating the certificate

You can use Digital Certificate Manager (DCM) to create a CIM server certificate that is issued by a CA on the operating system, or by an external CA.

Note: CIMOM is not integrated with DCM. All certificates that are created in DCM for CIMOM must be exported to CIMOM. CIMOM only supports the Privacy Enhanced Mail (PEM) format for certificates.

To create a certificate for CIMOM, follow these steps:

1. Create an application definition in DCM. The recommended application ID is QUME_CIMOM.
2. Create a certificate for the CIMOM application that is issued by a CA. Remember the subject name that you enter for CIMOM in the certificate.
3. Export the certificate from DCM to CIMOM.
   a. In the left frame, choose Manage Certificates and Export Certificates.
   b. Click **Server or client** as the type of the certificate.
   c. Select the certificate that you created for CIMOM and click **Export**.
   d. Click **File** as the export destination.
   e. Use the directory that is defined by the sslCertificateFilePath property for the export file name, and name the file `pegasuscert.p12`. This file is in PKCS12 (Public Key Cryptography Standards) format.
   f. Remember the password that you enter here. The password is used to decrypt the exported certificate.
4. Run the OpenSSL command to convert the certificate from the PKCS12 format to the PEM format.
   a. On the operating system, use the CALL QP2TERM command to make the i5/OS Portable Application Solutions Environment (i5/OS PASE) environment available.
   b. Change the directory to the location of the exported certificate.
   c. Extract the certificate from the PKCS12 file and convert it to the PEM format.

      Use the OpenSSL command: `OpenSSL pkcs12 -in pegasuscert.p12 -out pegasuscert.pem -nokeys -clcerts`. This command prompts for the password that you entered in the DCM export window.

      The PEM file might contain both the CIMOM certificate and the certificate of the CA that issues the CIMOM certificate. Because CIMOM does not support this type of PEM file, remove the CA certificate.
   d. Edit the PEM file and remove all the lines except the lines for the CIMOM certificate.

      The certificate has the CIMOM subject name that you used when creating the certificate in DCM. Keep the lines of CIMOM certificate starting with Bag Attributes and ending with End Certificate.
   e. Extract the private key from the PKCS12 file and convert it to the PEM format.

Use the following OpenSSL command: `OpenSSL pkcs12 -in pegasuscert.p12 -out pegasuskey.pem -nocerts -nodes`. This command prompts for the password that you entered in the DCM export window.

After you have the certificate and private key in the PEM format, you can make them available to CIMOM by placing them in the paths that are defined by the sslCertificateFilePath and sslKeyFilePath properties.

Note: When CIMOM starts, the private key file is created automatically. It is important to keep the private key in a protected directory. By default, the CIMOM private key is put in a directory that is owned by QSYS, with PUBLIC *EXCLUDE and no private authorities. If the administrator changes the sslKeyFilePath property, the new key directory should be protected.

After the SSL certificates are created, set the following configuration properties to enable the CIM server with SSL and disable the non-SSL port:

- enableHttpsConnection: set the value to true
- enableHttpConnection: set the value to false

Enabling the CIM server to verify client certificates

To enable SSL client certificate verification on the main SSL port, you can use the sslClientVerificationMode property. This property can be set to disable the client certificate verification, to require the client certificate verification, or to verify the client certificate if available and fall back to the httpAuthType property if the certificate is not available. With this property, you can be authenticated through certificate verification or basic authentication. The sslTrustStore property gives the location of the truststore. The exportSSLTrustStore property should have the same value as sslTrustStore; otherwise, the server cannot be started.

To enable SSL client certificate verification on the export SSL port, you can use the enableSSLExportClientVerification property. When the value is set to true, CIMOM requires the exported clients to send certificates. The exportSSLTrustStore property gives the location of the truststore. You need to restart the server to make the properties work after you change the value.

You can use the cimconfig command to set the current configuration properties or the planned configuration properties of the CIM server. You can change the following configuration properties:

- logdir
- logLevel
- shutdownTimeout
- traceComponents
- traceFilePath
- traceLevel

When you change the values of these properties, the changes take place immediately without restarting the server. You can update the current configuration properties only when the CIM server is running.

The properties that are not fixed can be changed in the planned configuration properties, whether the CIM server is running or stopped. If the planned configuration properties are changed, those changes do not take effect until the CIM server is restarted. When the CIM server is restarted, the planned configuration properties become the current configuration properties.

Related information

OpenSSL

Basic startup properties for CIMOM

You can change basic startup properties for CIMOM with the cimconfig command.

The following table describes the startup properties and default values of CIMOM. For these changes to take effect, you must restart the CIM server. The changes cannot be made dynamically.

Table 1. Basic startup properties

Property	**Default value**	**Description**
enableAuthentication	true	This property determines whether the program performs authentication before any request is allowed into the CIM server over the wbem-http and wbem-https ports. If the value is set to true, the program performs authentication before any request is allowed into the CIM server. This does not affect the wbem-exp-https port. If the value is set to false, the program allows unauthenticated requests access to the CIM server. If you need to set the value to false, ensure that the server environment is secure.
enableHttpConnection	true	This property controls whether to allow access through the HTTP port. If the value is set to true, you can access through the HTTP port. If you are sure that the server environment is unsecure, set the value to false. To ensure the security for the enableHttpConnection property, set the httpBindAddress property to 127.0.0.1 or ::1.
enableHttpsConnection	true	This property enables the HTTPS port to listen for HTTPS requests.
enableHttpExportConnection	true	This property enables the HttpExport port to listen for HTTP export requests.
httpAuthType	Basic	This property determines the authentication to be performed over the HTTP port. **Note:** Only basic authentication is supported. The Kerperos authentication is no longer supported.
httpBindAddress	127.0.0.1	This property determines which network interface to bind to for the HTTP port. To enable the server to bind to all available interfaces (IP address), set the value to ALL. For the value ALL, both IPv6 and IPv4 connections are accepted. If the value is a specific IP address, the enableHttpConnection property controls whether the CIM server listens for the specific address for an HTTP connection. If the IP address is set within the IPv6 family, the CIM server can only support an IPv6 connection. If the IP address is set within the IPv4 family, the CIM server can only support an IPv4 connection. By default, dual stacks are supported on the operating system. To disable the IPv6 stack, use the following commands: ENDTCP and STRTCP STRIP6(*NO).

Table 1. Basic startup properties (continued)

Property	Default value	Description
httpExportPort	""	This property specifies the port number of HTTP export requests that the server listens on. You should set the value to a valid port number. It overrides the port number of the wbem-http service in the TCP/IP services table. If the value is not set, the port from the wbem-http service is used. If neither this property nor the wbem-http service port is set, a default value 5988 is used. This property only takes effect if the enableHttpConnection property is set to true.
httpPort	""	This property specifies the port number of HTTP requests that server listens on. You should set the value to a valid port number. It overrides the port number of the wbem-http service in the TCP/IP services table. If the value is not set, the port from the wbem-http service is used. If neither this property nor the wbem-http service port is set, a default value 5988 is used. This property only takes effect if the enableHttpConnection property is set to true.
httpsBindAddress	ALL	This property determines which network interface (IP address) to bind to for the HTTPS port.
httpsPort	""	This property specifies the port number of HTTPS requests that server listens on. You should set the value to a valid port number. It overrides the port number of the wbem-https service in the TCP/IP services table. If the value is not set, the port from the wbem-https service is used. If neither this property nor the wbem-https service port is set, a default value of 5989 is used. This property only takes effect when the enableHttpConnection property is set to true.

Related concepts

"Authentication on CIMOM" on page 17
When a user request comes through HTTP or HTTPS, CIMOM determines whether this is a legitimate user on the system. If the request does not pass the authentication, the request is rejected. If you set the enableAuthentication property to false, the CIMOM authentication function is disabled.

Advanced startup properties for CIMOM

You can change the advanced startup properties for the CIM server with the cimconfig command. These properties are intended for use only by advanced CIMOM users.

The following table describes the advanced startup properties for the cimconfig command, their default values, and whether they can be changed dynamically (meaning that they take effect immediately without restarting the server).

Note: The shutdownTimeout, logdir, logLevel, traceLevel, traceComponents, and traceFilePath properties are dynamic. The other properties are not dynamic. For all the other properties, you must use the -p parameter to indicate your changes. You need to restart the CIM server to make the changes take effect.

Table 2. Advanced startup properties

Property	Dynamic	Default value	Description
crlStore	No	ssl/crlstore/	This property describes the path to the directory or file that contains the certificate revocation lists (CRLs). If this property is not set, set to an empty directory or set to an empty file so that no CRLs are loaded. This property only takes effect if the sslClientVerificationMode property is set to required or optional, or if the enableSSLExportClientVerification property is set to true.
enableAssociationTraversal	No	true	This property enables the association traversal. You can set the value to true or false.
enableIndicationService	No	true	This property enables the indication service. You can set the value to true or false.
enableNormalization	No	false	This property controls whether to normalize objects from trusted entities. This property only works in InProcess mode. This means that to make it take effect, you need to set the enableNormalization property to true and the forceProviderProcesses property to false. In OOP mode (the forceProviderProcesses property is set to true), the property does not work. This property has the following values: **true** The program ensures that the objects that are delivered from providers are complete and accurate. **false** The program does not normalize objects from trusted entities (for example, the objects from the repository, control providers, IBM-supplied providers, and certain vendor providers). It normalizes only the objects from the third-party providers that are added to a distribution.
enableSSLExportClientVerification	No	true	This property controls whether to allow exported clients to be connected using HTTPS on the port that is specified by the service name wbem-exp-https. Only CIM export requests are allowed on this port. You can set the value to true or false. **Note:** If the wbem-exp-https port is not defined in the system's TCP/IP services table, an error is recorded and the server does not start. The wbem-exp-https port is defined in the i5/OS services table by default.

Table 2. Advanced startup properties (continued)

Property	**Dynamic**	**Default value**	**Description**
enableSubscriptionsForNonprivilegedUsers	No	false	This property controls whether users need special authorities to create indication subscriptions. You can set the value to true or false. If the value is false, only a user with *IOSYSCFG and *ALLOBJ special authorities is allowed to create indication subscriptions.
excludeModulesFromNormalization	No	""	This property disables normalization for objects from specific provider modules. If the enableNormalization property is set to true, all provider objects are normalized except for those on this exclusion list.
exportSSLTrustStore	No	ssl/truststore/	This property specifies the path to the directory or file that contains the trusted certificates for CIM export requests. The truststore includes CA certificates. You must set this property if the enableSSLExportClientVerificationMode property is set to true. If this property is set to an empty directory or an empty file, no export certificates are trusted. This property only takes effect if the enableSSLExportClientVerification property is set to true. **Note:** The sslTrustStore and exportSSLTrustStore properties should be set to the same value.
forceProviderProcesses	No	true	This property controls how the providers run in processes. If you set the value to true, the providers run in separate processes rather than loading and calling provider libraries directly within the CIM server process.
Logdir	Yes	/QOpenSys/QIBM/UserData/UME/Pegasus/logs/	This property specifies the name of the directory that is used for the CIMOM-specific log files. **Note:** Make sure that the server has the authority to write files in the directory if you want to change this property.
logLevel	Yes	INFORMATION	This property sets the level of the data that is logged. The data is saved in the CIMOM log directory. The property has the following values: • FATAL • INFORMATION • SEVERE • TRACE • WARNING
maxProviderProcesses	No	0	This property limits the number of provider processes that run concurrently. A value of 0 indicates that the number of provider agent processes is not limited.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
messageDir	No	/QOpenSys/QIBM/ProdData/UME/Pegasus/msg	This property points to the default directory to search for the globalization message bundles. The default value points to the included message bundles.
providerDir	No	/QOpenSys/QIBM/ProdData/UME/Pegasus/provider	This property specifies the names of the directories that contain the providers that are running.
repositoryIsDefaultInstanceProvider	No	true	The property enables the repository component of the CIM server to provide CIM object instances by default. If the value is true, no providers service the client request for the CIM instance, and the CIM server repository is used. If the value is false, the i5/OS providers that implement CIM metric classes no longer function properly.
socketWriteTimeout	No	20	This property defines the timeout (in seconds) for the socket on the server.
sslCertificateFilePath	No	ssl/keystore/serverkey.pem	This property indicates the path to the certificate file of the CIM server. You must set this property to a valid certificate if the enableHttpsConnection or enableSSLExportClientVerification property is set to true. You can also set the sslCertificateFilePath property to a valid path. If no certificates are in the path, the server creates a certificate after the startup. **Note:** Certificates that are not valid and expired certificates are considered valid when they are loaded by the CIM server. A warning message is logged if the certificate is expired or is not valid. If the sslKeyFilePath property is not specified, the CIM server loads the private key from the certificate file.
sslKeyFilePath	No		This property indicates the path to the CIM server's private key file. If the certificate that is specified in the sslCertificateFilePath property contains the private key, you do not need to set this property. You need to keep this file in a protected directory as the default value.
sslTrustStore	No	ssl/truststore/	This property indicates the path to the directory or file that contains the trusted certificates for CIM operation requests. The truststore includes CA certificates. You must set this property if the sslClientVerificationMode property is set to required. If the sslClientVerificationMode property is set to optional, this property should be set to empty. In this case, no certificates are trusted. If the sslClientVerificationMode is set to disabled, this property is not used.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
sslClientVerificationMode	No	optional	The property sets the mode of SSL client certificate verification. This property is only effective if the enableHttpsConnection property is set to true. You can set the property to the following values: **required** The CIM server requires the verification of a client certificate on the HTTPS port and rejects the request if the client certificate is not trusted. The httpAuthType property is not used. **optional** The CIM server verifies a client certificate if available; otherwise, the CIM server uses the httpAuthType setting for client verification. **disabled** The CIM server uses the httpAuthType setting for client verification.
sslTrustStoreUserName	No	""	This property identifies the user name as the user context for the CIM operation request when certificate authentication is used and a user name cannot be associated with a specific certificate file. The user context is the i5/OS user profile under which the provider is called to perform the CIM request. You must set this property to a valid user profile on the operating system. If the sslClientVerificationMode property is set to disabled, this property is not effective. If the sslTrustStore property is set to a directory, this property is not effective. If the sslTrustStore property is set to a single file, this property must be set to a username; otherwise, an error is reported and the CIM server does not start. In this case, all the certificates in the file are assigned to the username that is specified by the sslTrustStoreUserName property.
shutdownTimeout	Yes	10	This property specifies the maximum number of seconds allowed for the CIM server to complete requests before it shuts down. When the ENDTCPSVR *CIMOM command is issued, the timeout is the maximum number of seconds that are allowed for the CIM server to complete outstanding CIM operation requests before it shuts down. If the specified timeout period expires, the CIM server shuts down, regardless of whether CIM operations are in progress. The minimum value of this property is 2.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
traceFilePath	Yes	/tmp/cimserver.trc	This property indicates the path to the trace file. The trace of the CIM server is written in /tmp/cimserver.trc. The trace of Well out-of-process providers is written in their own trace files with prefix cimserver.trc and suffix provider module.user (for example, cimserver.trc.TestProviderModule.qycmcimom). **Note:** Make sure that the server has the authority to write files in the directory if you want to change the property.
traceComponents	Yes	""	This property specifies the components that you want to trace in CIMOM. The valid settings are listed in "Settings for the traceComponents property."
traceLevel	Yes	1	This property indicates the level of debug trace. If the value is 1, the program only traces function exits (the minimum trace). A trace level of 4 is the maximum trace. **Note:** If the traceLevel property is set to 4 and the traceComponents property is set to ALL, the size of trace file grows quickly and uses large disk spaces. The range of value is 1 to 4.
idleSessionTimeout	No	0	This property indicates the minimum timeout value for idle client connections. If the value is 0, idle client connections are not disconnected.

Related concepts

"Authentication on CIMOM" on page 17
When a user request comes through HTTP or HTTPS, CIMOM determines whether this is a legitimate user on the system. If the request does not pass the authentication, the request is rejected. If you set the enableAuthentication property to false, the CIMOM authentication function is disabled.

"User authorization on CIMOM" on page 17
User authorization is a type of security check that verifies whether you have access to the objects you want to change. Authorization is needed not only for changing operations but sometimes for reading operations as well.

Settings for the traceComponents property

You can use the traceComponents property to trace components. This topic lists the valid settings for the traceComponents property.

- ALL
- AsyncOpNode
- Authentication
- Authorization
- BinaryMessageHandler
- Channel
- CimData
- CIMExportRequestDispatcher
- CIMOMHandle

- 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
- SubscriptionService
- Thread
- UserManager
- WQL
- XmlIO
- XmlParser
- XmlReader
- XmlWriter

Tracing is disabled by default and should be used for debugging purposes. You can enable the tracing mechanism by specifying the trace level and the component that you want to trace with the traceLevel property. Possible trace levels follow:

Level 1
Function entry and exit

Level 2
Basic flow, trace messages, and low-data detail

Level 3
Interfunction logic flow, medium-data detail

Level 4
All information, high-data detail

The trace data is saved in the file that is specified by the traceFilePath property. By default, the traceFilePath property is set to /tmp/cimserver.trc.

You can also trace all the components by setting ALL where you specify components in the property. If the traceComponents property is not set to any component, tracing is disabled regardless of the setting of the traceLevel property.

You can use the cimconfig command to modify the trace configuration parameters when 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, open an i5/OS PASE shell and complete these steps:

1. Call qp2term.
2. Type the following commands:
   a. **cimconfig -s traceLevel=4**
   b. **cimconfig -s traceComponents=Thread, ProvManager**

Similarly, to disable all tracing, type the following command: **cimconfig -s traceComponents=**

Starting and stopping CIMOM

You can start or stop a CIMOM job from System i Navigator, from IBM Systems Director Navigator for i5/OS, or from a command-line interface.

About this task

- To start or stop a CIMOM job from System i Navigator, follow these steps:
  1. From System i Navigator, select **Network** → **Servers** → **User-Defined**.
  2. Select **CIMOM**. You can use this window to start or stop CIMOM, and to determine whether CIMOM starts with TCP/IP by default.
- To start or stop a CIMOM job from IBM Systems Director Navigator for i5/OS, follow these steps:
  1. Log on to *your system*.
  2. Click **i5/OS Management** → **Network**.
  3. In the Task field, select **Show All Network task**.
  4. Select **User-Defined** and click **OK**. On the User-Defined page, you can work with CIMOM.
- To start or stop a CIMOM job from a command-line interface, use the following commands:
  – To start the CIMOM job, use the STRTCPSVR *CIMOM command.
  – To end the CIMOM job, use the ENDTCPSVR *CIMOM command.

CIMOM security

You have several options to ensure that the CIM server is secure on the i5/OS operating system. In the Common Information Model Object Manager (CIMOM), two types of security checks are available: authentication and authorization.

Related concepts

"cimconfig usage information" on page 25
You can configure the startup properties for CIMOM with the cimconfig command.

Network authentication service

Host name resolutions considerations

Authentication on CIMOM

When a user request comes through HTTP or HTTPS, CIMOM determines whether this is a legitimate user on the system. If the request does not pass the authentication, the request is rejected. If you set the enableAuthentication property to false, the CIMOM authentication function is disabled.

Local users are users on a system who are sending requests to CIMOM on the same system. *Remote users* are users on a system who are sending requests to CIMOM on another system. By default, CIMOM uses Secure Sockets Layer (SSL) for all remote communications, with client-side and server-side certificates that are trusted by the management applications.

Local user authentication

For local users, CIMOM uses a local authentication mechanism. CIMOM accepts the authentication that is already done by the system itself so that local requests include only the users' login names without their passwords. HTTP authentication is still used, but because the user is already logged in, no password is needed.

Remote user authentication

Remote users are authenticated by HTTP basic authentication or HTTPS SSL peer certificate authentication. Configuration settings for the following properties determine which mechanisms are used:

- enableAuthentication
- enableHttpConnection
- enableHttpsConnection
- enableSSLExportClientVerification
- httpAuthType
- sslClientVerificationMode

For detailed information about the descriptions and default values of these properties, see the information about basic and advanced startup properties.

Related concepts

"Basic startup properties for CIMOM" on page 7
You can change basic startup properties for CIMOM with the cimconfig command.

"Advanced startup properties for CIMOM" on page 9
You can change the advanced startup properties for the CIM server with the cimconfig command. These properties are intended for use only by advanced CIMOM users.

"Object authorities" on page 20
IBM Universal Manageability Enablement products are installed in the UserData and ProdData directories in the integrated file system and QUME library in the library file system. You need certain authorities to access these directories and this library.

Authorization on CIMOM

Authorization on CIMOM includes user authorization, command authorization, and object authorities.

User authorization on CIMOM

User authorization is a type of security check that verifies whether you have access to the objects you want to change. Authorization is needed not only for changing operations but sometimes for reading operations as well.

The CIM operations can be divided into two kinds: operations that access the repository files that are owned by the CIM server, and operations that call the provider exit programs to manage system resources.

Authorization to CIM class and qualifier operations

CIM class and qualifier operations change the local copy of the CIM schema. You must have the authority to these operations before you can perform these operations with systems management data that is provided by CIM. These operations do not change any i5/OS system objects, but they change the CIM schema. For the i5/OS operating system, Application Administration in System i Navigator controls authorization to these operations.

Provider user context

Some CIM providers run as exit programs to the server. Providers dynamically load and call the CIM server to perform CIM operations. They are plug-ins and run in i5/OS Portable Application Solutions Environment (i5/OS PASE).

The CIM server needs to run under QSECOFR authority to switch the user profile under which the providers are running. The providers can be run under the following user profiles:

- The profile of the requesting client.
- The profile of the CIM server.
- A designated profile.
- The root authority that is the QUMECIMOM object on the operating system.

For more information, refer to the forceProviderProcesses property in the Advanced startup properties for CIMOM topic.

To set the user profile for the provider, the CIM server creates a new job for the provider, sets the user profile of that job, and runs the provider in that job.

For security considerations, the CIM server writes an audit journal for security events, such as password check failure and special-authority check failure. An audit journal entry is created for each failure.

Related concepts

"i5/OS metrics classes" on page 95
The topic describes metric classes and the user authorization.

"Advanced startup properties for CIMOM" on page 9
You can change the advanced startup properties for the CIM server with the cimconfig command. These properties are intended for use only by advanced CIMOM users.

Working with authorization for CIM operations:

You can use Application Administration to work with the authorization for CIM operations.

About this task

To work with the authorization for CIM operations, follow these steps:

1. From System i Navigator, expand *your system* → **Application Administration**.
2. Select **Local Settings**, if available.
3. Select **Host Applications** → **CIMOM server**.
4. Add or remove a user or group's authorization to the following operations.
   - CreateClass
   - DeleteClass

- DeleteQualifier
- EnumerateClasses
- EnumerateClassNames
- EnumerateQualifiers
- GetClass
- GetQualifier
- ModifyClass
- SetQualifier

Results

Note: If a user wants to write a CIMOM private namespace (PG_InterOp, PG_Internal), you need to check the setting of Application Administration and check whether the user have *ALLOBJ *IOCFG authority. For more information about the namespaces that are installed with CIMOM, see the information about backup and recovery considerations.

Related concepts

"Backup and recovery considerations for CIMOM" on page 23
It is important to schedule backups of the repository directories and files. If the repository is moved, is lost, or becomes corrupted, restore the files that you have backed up.

Working with authorization for CIM metrics classes:

You can use Application Administration to work with the authorization for CIM metrics classes.

About this task

To work with authorization for CIM metrics classes, follow these steps:

1. From System i Navigator, expand *your system* → **Application Administration**.
2. Select **Local Settings**, if available.
3. Select **Host Applications** → **CIMOM server** → **System Management Operations** → **Access to the CIM Performance Provider**.
4. Set the default authorization or add or remove a user or group's authorization.

Command authorization

You need to use commands to modify the configuration of the CIM server. Command authorization is required.

External command-line interfaces have PUBLIC *X (UNIX® 701 permissions) and have a symbolic link in the /QOpenSys/usr/bin directory.

Most command-line interfaces act as a CIM client and communicate with the server through a local connection. These clients authenticate to the server using the local authentication and then send the CIM operation to the CIM server. The server authorizes the client to do the CIM operation by checking whether the authenticated user has special authorities (for example, *IOSYSCFG and *ALLOBJ). For the command-line interfaces that are not CIM clients, the special authority check is done in the command-line interface itself.

The cimmof command is an exception to this rule. This command uses the CIM client to modify the repository files of the CIM server. Local authentication is used, but the server authorizes the cimmof command caller to modify the repository based on the Application Administration settings. You do not need to add special authority checks for these repository requests in addition to the Application Administration checks.

Object authorities

IBM Universal Manageability Enablement products are installed in the UserData and ProdData directories in the integrated file system and QUME library in the library file system. You need certain authorities to access these directories and this library.

Protection of the UserData directories

- Base directory: /QOpenSys/QIBM/UserData/UME/Pegasus/
- CIM repository:
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/root
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#cimv2
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#ibmsd
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#PG_Internal
  - /QOpenSys/QIBM/UserData/UME/Pegasus/repository/root#PG_InterOp
- SSL stores:
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/crlstore
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/exporttruststore
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/keystore
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ssl/truststore
- Director mappings:
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Events
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Events/data
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Events/logs
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mif
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mif/data
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/sql
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/sql/data

The base directory, /QOpenSys/QIBM/UserData/UME/Pegasus, is owned by QSYS. Its access mode is PUBLIC *RX, which allows access to these directories and files through the CIMOM command-line interface and CIM request interfaces.

Protection of the ProdData directories

- Libraries: /QOpenSys/QIBM/ProdData/UME/Pegasus/lib/
- Programs: /QOpenSys/QIBM/ProdData/UME/Pegasus/bin/
- IBM-supplied providers: /QOpenSys/QIBM/ProdData/UME/Pegasus/provider
- Messages:
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/msg/pegasus
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/msg/provider
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/msg/ibm
- Schemas:
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Schemas/CIM
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Schemas/Pegasus/Internal
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Schemas/Pegasus/InterOp

  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Schemas/Pegasus/ManagedSystem
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Schemas/OS400
- Mappings directories:
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Events
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Events/bin
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Events/data
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Events/lib
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mib
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mib/bin
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mib/data
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mif
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mif/bin
  - /QOpenSys/QIBM/ProdData/UME/Pegasus/Mappings/Inventory/mif/data
- ICU Libraries: /QOpenSys/QIBM/ProdData/UME/ICU/icu-3.4.0/lib/

All of the directories and files in the directories are owned by the QSYS property; their access mode is PUBLIC *RX.

The following directories have internal server files:
- /QOpenSys/QIBM/ProdData/UME/Pegasus/bin/
- /QOpenSys/QIBM/ProdData/UME/Pegasus/lib/
- /QOpenSys/QIBM/ProdData/UME/Pegasus/provider/

The access mode of the /lib/ directory is PUBLIC *RX. Only the external files have access mode PUBLIC *X. No files set the setuid bit.

The access mode of the provider/ directory is PUBLIC *RX. This directory only contains the IBM-included providers.

Note: You can modify the configuration of the providerDir property to make the CIM server load the providers from the directory that is created by the providers. The created directory should have permission PUBLIC *RX so that the providers can be loaded and run by any user.

The access mode of the msg/ directory is PUBLIC *RX. All files have permission PUBLIC *R because providers are loading messages under the user authority.

The Schemas/ directory and all files have permission PUBLIC *RX. These are source files.

The Mappings/ directory and all files have permission PUBLIC *RX.

Protection of objects in the QUME library

Table 3. Library objects

Library objects	Object type	Public authentication	Description
QUME	*LIB	*USE	Principle licensed program (LP) library
QUME/QCIMMSG	*MSGF	*USE	Message file
QUME/QUME0029	*PRDLOD	*USE	Machine readable information (MRI) product load

Table 3. Library objects (continued)

Library objects	Object type	Public authentication	Description
QUME/QUME0050	*PRDDFN	*USE	Product definition
QUME/QUME0050	*PRDLOD	*USE	Machine readable material (MRM) product load
QUME/QUME2MIB	*PGM	*EXCLUDE	SNMP subagent
QUME/QUMECIMOM	*PGM	*EXCLUDE	Wrapper to start CIMOM
QUME/QUMECIMV2	*FILE	*EXCLUDE	root/cimv2
QUME/QUMECTLCIM	*PGM	*EXCLUDE	CIMOM control program that is called by TOC component during STR/ENDTCPSVR *CIMOM
QUME/QUMEIBMSD	*FILE	*EXCLUDE	root/ibmsd
QUME/QUMEJOBD	*JOBD	*EXCLUDE	Job description for server and SNMP subagent
QUME/QUMEMRIPGM	*PGM	*EXCLUDE	MRI installation exit program
QUME/QUMEMRMPGM	*PGM	*EXCLUDE	MRM installation exit program
QUME/QUMEPGINOP	*FILE	*EXCLUDE	root/PG_InterOp
QUME/QUMEPGINTL	*FILE	*EXCLUDE	root/PG_Internal
QUME/QUMEPGROOT	*FILE	*EXCLUDE	root
QUME/QUMERECOVR	*PGM	*EXCLUDE	Recovery program
QUME/QUMESPSNDR	*SRVPGM	*USE	Check SNMP trap
QUME/QUMESTRSA	*PGM	*USE	Start or stop Simple Network Management Protocol (SNMP) subagent
QUME/QUMEUTIL	*SRVPGM	*USE	Audit log utility, *USE
QUME/QUMEUTILS	*SRVPGM	*EXCLUDE	Platform utilities (used by QUME2MIB and QUMESTRSA)

Adopted owner authority

The only objects that adopt owner authority are QUMECTLCIM *PGM and QUMESTRSA *PGM. The owner of these programs is the QSYS object. The programs start and stop the CIM server or SNMP subagent.

QUMECTLCIM is the program that starts and stops the CIM server. This program has PUBLIC *EXCLUDE authority. QUMECTLCIM starts the server by submitting the QUMECIMOM server job. QUMECTLCIM is called by QTOCSRV and adopts the QSYS object owner authority to gain access to the job description of the CIM server (QUMEJOBD). The QUMEJOBD job description sets the user of the QUMECIMOM server job to QSECOFR.

Similarly, the QUMESTRSA object is the *PGM that starts and stops the SNMP subagent. This program has PUBLIC *USE authority so that it can be called by any user. The QUMESTRSA object starts the subagent by submitting the QUME2MIB job. It adopts the QSYS object owner authority to gain access to the job description of the subagent (QUMEJOBD). The QUMEJOBD job description sets the user of the QUME2MIB server job to the QSECOFR object.

Related concepts

"User authorization on CIMOM" on page 17
User authorization is a type of security check that verifies whether you have access to the objects you want to change. Authorization is needed not only for changing operations but sometimes for reading operations as well.

Related reference

"Command authorization" on page 19
You need to use commands to modify the configuration of the CIM server. Command authorization is required.

Backup and recovery considerations for CIMOM

It is important to schedule backups of the repository directories and files. If the repository is moved, is lost, or becomes corrupted, restore the files that you have backed up.

Here are the namespaces that are installed with CIMOM:

root The root namespace conforms to the Distributed Management Task Force (DMTF) specifications.

root/cimv2
This is for standard CIM schemas for the shipped providers.

root/PG_InterOp
This is for provider registration. This space is reserved exclusively for providers and all providers must be registered.

root/PG_Internal
This space is reserved and used by CIMOM.

root/ibmsd
The namespace is owned and used by IBM Director.

Backup

IBM data and User data are stored in the CIM server UserData/ directory. You need to back up this directory daily because applications or providers might change the files in the directory. You can run the **backup** command to save the changed files since the last backup.

Here are more details about the specific files and directories:

- The repository as a whole can be backed up, including any temporary transaction files.
- The server configuration file (current and planned) can be backed up.
- The configuration files are located under the /QOpenSys/QIBM/UserData/UME/Pegasus/ directory.
- The SSL files in the UserData/ directory are not IBM files. These files can be backed up, including the server certificate, the private key, and trust stores. However, the backup location should be secure.
- The mappings files in the UserData/ directory are not IBM files. These files can be backed up.
- The migration marker file is a user file that can be backed up.
- The trace files and log files are user files that can be backed up.

The Universal Manageability Enablement program has product files in its Proddata directory and includes the QUME product library. These files are backed up weekly.

Notes:

- It is important that you back up the /QOpenSys/QIBM/UserData/UME/Pegasus/repository directory structure regularly. If these files are deleted, moved, or corrupted, you need to restore them from the backup.

- To back up the SSL certificate files for CIMOM, use the OpenSSL command to create the certificates. For more detailed information about the command, see OpenSSL .
- If you do not back up the repository, you need to delete the repository and restart the CIM server. This sets the repository to the default state.

Recovery

In CIMOM, IBM data and user data is intermingled as files or as data in the same directory.

The following files contain both IBM and user data:

- All instance index files and instance data files. A typical example is the provider registration that contains both IBM and user data instances. Any providers that are developed by you are registered in the same files as the IBM-supplied providers.
- Instance association and class association files. These files keep track of associated classes or instances. Any associations that are created by you are stored in the same file as the IBM-supplied associations.
- Server configuration files.

If you need to recover the ProdData/ directory of the CIM server, reinstall the QUME product library. If you need to recover the IBM data in the UserData/ directory of the CIM server, reinstall the QUME library or recover these files from the backup media. If you need to recover the user data in the UserData/ directory of the CIM server, recover the destroyed files from the backup media.

Related tasks

"Working with authorization for CIM operations" on page 18
You can use Application Administration to work with the authorization for CIM operations.

Related information

Backing up your system

Restoring corrupted files

Use this information if the backup copy of your CIM repository files are corrupted.

About this task

To recover your files, use the information in the following list.

Repository classes and qualifiers (static data)

1. Undo whatever was done to create the class or qualifier. For example, uninstall a client application or take manual steps to undo what was done.
2. Put the class or qualifier back the same way it was before. For example, reinstall a client application. If the problem persists, contact your service provider.

Repository instances

1. Undo whatever was done to create the instance. For example, uninstall a client application or take manual steps to undo what was done.
2. Put the class or qualifier back the same way it was before. For example, reinstall a client application. If the problem still exists, contact your service provider.

Provider registration data (also instances)

1. Use the cimprovider command to remove the provider registration.
2. Use the cimmof command to recompile and reregister the data. If the problem still exists, contact your service provider.

CIMOM command-line utilities

You can use a set of command-line utilities to control or change the CIMOM environment. These command-line utilities include cimmof, cimconfig, cimprovider, and ssltrustmgr.

For the i5/OS implementation, the cimconfig and cimprovider commands require *IOSYSCFG and *ALLOBJ special authorities. The ssltrustmgr command requires *ALLOBJ and *SECADM special authorities. You do not need special authorities to run the cimmof command.

In CIMOM, you need to comply with the namespaces and with the authority checks that are based on the objects.

Run all of the command-line utilities from a command line or in i5/OS Portable Application Solutions Environment (i5/OS PASE).

Related tasks

"Configuring CIMOM" on page 4
When you install the i5/OS operating system, the IBM Universal Manageability Enablement for i5/OS licensed program is installed on the system by default. Before you use the CIM function of this licensed program, you need to configure the Common Information Model Object Manager (CIMOM).

cimconfig usage information

You can configure the startup properties for CIMOM with the cimconfig command.

You can use this command to update configuration setting. A symbolic link in the /QOpenSys/usr/bin directory for this command is provided. If you change the configuration properties that are in the planned configuration settings, the changes do not take effect until the CIM server is restarted.

Name `cimconfig`

Get, set, unset, or list CIMOM configuration properties.

Synopsis

Usage:

- `cimconfig -g name [ -c ] [ -d ] [ -p ]`
- `cimconfig -s name=value [ -c ] [ -p ]`
- `cimconfig -l [ -c | -p ]`
- `cimconfig -u name [ -c ] [ -p ]`
- `cimconfig -h`
- `cimconfig --help`
- `cimconfig --version`

Remarks

The cimconfig command provides a command-line interface to manage CIMOM configuration properties:

- The first form of cimconfig provides the current, planned, and default value of the specified configuration property.
- The second form sets the current value and planned value of the specified configuration property to the specified value.
- The third form lists all the configuration properties.
- The fourth form resets the current and planned values of the specified configuration property to its default value.

Options

The cimconfig command recognizes the following options:

-h, --help
This option displays command help information.

--version
This option displays the CIMOM version.

-g name
This option gets the current value of the specified configuration property. It returns an error when CIMOM is not running.

-g name -c
This option gets the current value of the specified configuration property. It returns an error when CIMOM is not running.

-g name -p
This option gets the planned value of the specified configuration property.

-g name -d
This option gets the default value of the specified configuration property. It returns an error when CIMOM is not running.

-s name=value
This option indicates that a configuration property is added or updated by setting its current value to the specified value. It returns an error when CIMOM is not running or when the specified property cannot be updated dynamically.

-s name=value -c
This option indicates that a configuration property is added or updated by setting its current value to the specified value. It returns an error when CIMOM is not running or when the specified property cannot be updated dynamically.

-s name=value -p
This option indicates that a configuration property is added or updated by setting its planned value to the specified value.

-u name
This option indicates that the current value of the specified configuration property is reset to the default value. It returns an error when CIMOM is not running or when the specified property cannot be updated dynamically.

-u name -c
This option indicates that the current value of the specified configuration property is reset to the default value. It returns an error when CIMOM is not running or when the specified property cannot be updated dynamically.

-u name -p
This option indicates that the planned value of the specified configuration property is reset to the default value.

-l This option displays the name of all the configuration properties. It returns an error when CIMOM is not running.

-l -c This option displays the name-and-value pair of all the current configuration properties. It returns an error when CIMOM is not running.

-l -p This option displays the name-and-value pair of all the planned configuration properties.

Note: You can use the cimconfig command to set the current or planned configuration properties of CIMOM. You can update the current configuration properties only when CIMOM is running. All of the properties can be changed in the planned configuration properties whether or not CIMOM is running. If the planned configuration properties are changed, those changes do not take effect until CIMOM is restarted. When CIMOM is started, the planned configuration properties become the current configuration properties.

Related concepts

"CIMOM security" on page 16
You have several options to ensure that the CIM server is secure on the i5/OS operating system. In the Common Information Model Object Manager (CIMOM), two types of security checks are available: authentication and authorization.

Related tasks

"Configuring CIMOM" on page 4
When you install the i5/OS operating system, the IBM Universal Manageability Enablement for i5/OS licensed program is installed on the system by default. Before you use the CIM function of this licensed program, you need to configure the Common Information Model Object Manager (CIMOM).

cimmof usage information

You can use this command to compile Managed Object Format (MOF) files. A symbolic link in the /QOpenSys/usr/bin directory for this command is provided.

Name `cimmof`

This command compiles CIM class description (using the MOF language) files into a class schema that is stored in a repository through the CIM server. This command only works when the CIM server is running.

Synopsis

Usage:

```
cimmof -h | --help
```

```
cimmof --version
```

```
cimmof [ -w ] [ -E ] [ -uc ] [ -aE | -aV | -aEV ] [ -I path ][ -n namespace ]
[--namespace namespace] [--xml][--trace] [mof_file ...]
```

Description

The cimmof command is the command line interface to the MOF compiler. The MOF compiler is a utility that compiles MOF files into CIM classes and instances that are stored in the CIM repository. You can use this command to compile MOF files at any time after installation. If no input file is specified, standard input is used. You need to provide the MOF file name in the message that is shown.

The MOF compiler requires that the input MOF files exist 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 that contain a list of MOF files using the following include pragmas:

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

MOF files that use the include pragma must be in the current directory or in a directory specified by the -I command line option. The -n option can be used to specify an R namespace in which the CIM classes and instances are compiled. If this option is not specified, the default R namespace is root/cimv2.

Options

-h, --help
This option displays command usage information.

--version
This option displays CIM server version.

-E This option performs a syntax check on the input. This option does not update the repository.

-w This option suppresses warning messages. If the CIM elements (such as classes, instances, properties, or methods) that are defined in the MOF files exist in the CIM repository, the cimmof command returns warning messages.

-uc This option allows the update of an existing class definition. This option enables you to update a leaf class. It does not allow updates of superclasses or classes that have subclasses.

-aE This option allows experimental schema changes.

-aV This option updates a class that results in a version change. This option allows the major version of the class to be changed, allows the version to be degraded or allows the version to be removed. The version must be specified in a valid format. The format is *m.n.u* where *m* is a major version, *n* is a minor release, and *u* is an update. For example, 2.7.0 is a valid format for CIM schema 2.7.0. If the input class has the same version as the class in the repository, the class is not updated.

-aEV This option allows both experimental and version schema changes.

-I <path>
This option specifies the path to include MOF files. This path might be relative or absolute.

-n This option overrides the default CIM repository namespace. The namespace that is specified must be a valid CIM namespace name. For provider registration schemas, the namespace that is specified must be root/PG_InterOp.

--namespace
This option overrides the default CIM repository namespace. The namespace that is specified must be a valid CIM namespace name. For provider registration schemas, the namespace that is specified must be root/PG_InterOp.

--xml This option generates Extensible Markup Language (XML) to a standard output format. This option does not update the repository.

--trace This option writes the trace information to a file. The output destination is a standard output format.

EXIT STATUS
The cimmof command returns one of the following values:

0 Success

1 Error

Examples

`cimmof processInfo.mof`
It compiles an MOF file into the default namespace in the CIM repository and issues the cimmof command with no options.

`cimmof -n root/application test1.mof test2.mof`
It compiles the MOF files into the root/application namespace.

`cimmof -w -I. /MOF MOF/CIMSchema25.mof`
It compiles the MOF file that is defined in the . /MOF directory with the name CIM-Schema25.mof and that contains include pragmas for other MOF files also in the . /MOF directory.

`cimmof -h`
It displays usage information for the cimmof command.

cimprovider usage information

You can use this command to enable or disable a registered provider. A symbolic link in the /QOpenSys/usr/bin directory for this command is provided. You need to ensure that CIMOM is running when you use this command.

Name `cimprovider`

Disable, enable, remove, or list registered CIM providers or one CIM provider module and module status.

Synopsis

Usage:

- `cimprovider -d -m module`
- `cimprovider -e -m module`
- `cimprovider -r -m module [ -p provider ]`
- `cimprovider -l [ -s | -m module ]`
- `cimprovider -h`
- `cimprovider --help`

Limitations

This command disables, enables, or removes only one CIM provider module or CIM provider at a time.

Description

If a CIM provider is disabled, CIMOM rejects any requests to the provider. If a CIM provider is enabled, CIMOM forwards requests to the provider. If a CIM provider is unregistered, CIMOM no longer has any information about the provider:

- The first form of the cimprovider command disables the specified provider module. When a specified provider module is in the disabled state, any new requests to the providers that are contained in the specified provider module are rejected.
- The second form enables the providers that are contained in the specified provider module. The providers are now ready to accept new requests.
- The third form removes the specified provider module and all of its providers, or removes the specified provider in the specified provider module.
- The fourth form lists all the registered provider modules and module status, or lists the providers in the specified provider module.

Options

-h, --help
The option displays command help information.

--version
The option displays the CIMOM version.

-d The option disables the specified CIM provider module. If the module is already disabled, an error message is returned.

-e The option enables the specified CIM provider module. If the module is already enabled or is currently being disabled, an error message is returned.

-r The option removes the specified provider module and all of its contained providers. If a provider is specified, it removes the specified provider in the specified provider module without affecting any other providers in that module.

-l The option displays all the registered provider modules.

-m Module
: The option specifies the provider module for the operation.

-p Provider
: The option specifies the provider for the operation.

-s The option displays the status of provider modules.

Examples

`cimprovider -d -m myProviderModule`
: It disables provider module `myProviderModule` and all of its contained providers (placing them in a stopped state).

`cimprovider -e -m myProviderModule`
: It enables provider module `myProviderModule` and all of its contained providers (placing them in an OK state).

`cimprovider -r -m myProviderModule`
: It removes (unregisters) the `myProviderModule` provider module and all of its contained providers.

`cimprovider -r -m myProviderModule -p MyProvider`
: It removes the `MyProvider` provider that is contained in the `myProviderModule` provider module.

`cimprovider -l`
: It lists the registered provider modules.

`cimprovider -l -s`
: It lists the registered provider modules and their status (such as OK, Stopping, Stopped).

`cimprovider -l -m myProvider`
: It lists the registered providers that are in the `myProviderModule` provider module.

ssltrustmgr usage information

The ssltrustmgr command provides a command-line interface to manage X.509 certificates in a truststore or in a certificate revocation list (CRL).

To run the ssltrustmgr command from an i5/OS PASE command line, you must install the i5/OS Portable Application Solutions Environment (i5/OS PASE) feature on the system. Make sure that the CIM server is running when you run this command from the /QOpenSys/usr/bin directory.

Name `ssltrustmgr`

This command requires *ALLOBJ and *SECADM authorities to display or change the SSL truststores.

Synopsis

Usage:

```
ssltrustmgr -a -c certuser -f certfile

ssltrustmgr -a -f certfile

ssltrustmgr -a -R -f crlfile

ssltrustmgr -h | --help

ssltrustmgr -l [-i issuername [-n serialnumber]]

ssltrustmgr -l -R [-i issuername]

ssltrustmgr -r -i issuername –n serialnumber

ssltrustmgr -r -R -i issuername

ssltrustmgr -v | --version
```

Description

If the truststore or the CRL store does not exist, or if they are not in a directory format, the command exits with errors.

Options

-a This option adds the specified certificate to a target truststore, a trustpath, or a CRL store.

-r This option removes the certificate that matches the serial number that is issued by the issuer name from the target truststore or trustpath.

-l This option displays the X.509 certificates in the target truststore or trustpath.

-R This option indicates that the requested add, remove, or list operation is performed on the CRL store.

-f certfile/crlfile
This option specifies a Privacy Enhanced Mail (PEM) format file that contains an X.509 certificate or a CRL.

-c certuser
This option specifies a user name to be associated with the specified certificate. The user name that is specified should be a valid system user on the target system.

-i issuername
This option specifies a certificate or a CRL issuer name.

-n serialnumber
This option specifies a certificate serial number.

-h | --help
This option displays the command help message.

-v | --version
This option displays the CIMOM number.

EXIT STATUS
When an error occurs, an error message is written to a standard error format, and an error value of 1 is returned.

0 Success

1 Error

Examples

`ssltrustmgr -a -c john -f cert.pem`
It adds the X.509 certificate in the cert.pem file to the truststore on CIMOM, and associates the user tag with the certificate.

`ssltrustmgr -a -R -f class1crl.pem or ssltrustmgr -aR -f class1crl.pem`
It adds the CRL in class1crl.pem to the CRL on CIMOM.

`ssltrustmgr -r -i ″/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com″ -n 01`
It removes the certificate that matches the specified issuer name and serial number from the cim_trust trust store.

`ssltrustmgr -l`
It lists all the X.509 certificates in the trust store.

`ssltrustmgr -lR -i ″/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com″`
It lists the CRL that is issued by the issuer name: /C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com

Dependencies for SNMP and SLP

Here are the dependency considerations for Simple Network Management Protocol (SNMP) and Service Location Protocol (SLP) supports.

SNMP support

The IBM Universal Manageability Enablement for i5/OS licensed program provides an SNMP subagent as a bridge between CIMOM and the SNMP server.

To start the SNMP subagent, use the `call qume/qumestrsa` command.

To stop the SNMP subagent, use the `call qume/qumestrsa stop` command.

When the SNMP subagent starts up, it registers itself to the SNMP server. The SNMP server forwards the request from the SNMP client to the subagent. After the subagent receives the SNMP request, the subagent can translate it to the CIM operation. Then the SNMP subagent is connected to the CIM server through the CIM client and sends the CIM operation.

SLP support

The CIM server supports self-registration with the SLP service agent. Both the IPv4 and IPv6 wildcard addresses are available on the CIM server. The server binds to more than one network interface, so the registrations in the Service Agent contain multiple entries. The registration contains a URL that locates the CIM server.

IPv4 example

```
URL: service:wbem:https://9.186.110.61:5989 ATTR: (template-url-syntax=service:wbem:https://
9.186.110.61:5989)
```

IPv6 example

```
URL: service:wbem:https://[3FFE:1::130]:5989 ATTR: (template-url-syntax=service:wbem:https://
[3FFE:1::130]:5989)
```

Reference information for CIM

The Common Information Model (CIM) standard provides the ability to develop management application that work with the systems management data that is made available by the CIM providers and included with the operating system.

The following Common Information Model Object Manager (CIMOM) functions are supported:

- **Large address-space models**

  The IBM Universal Manageability Enablement licensed program can use a maximum of 8 segments or 2 GB memory.
- **Secure Sockets Layer support**

  Secure Sockets Layer (SSL) is supported both for external connections over SSL-secured ports for CIM client connections and for the connections with a CIM export client. CIMOM supports the following ports:
  - wbem-https port (5989 by default)
  - wbem-exp-https port (5990 by default)

  Note: The original wbem-http port (5988 by default) is also supported.

In addition to the support for SSL-secured data transmission, i5/OS Portable Application Solutions Environment (i5/OS PASE) also supports SSL certificate-based client authentication on CIM requests from CIM clients and supports the CIM exports carrying indication data. You can configure SSL certificate-based client authentication as follows:

  – When the exclusive type of authentication mechanism is used (basic authentication is not used), only SSL certificate-based authentication is allowed. SSL data encryption is used.
  – The client is requested to provide a certificate; if one is not provided, the httpAuthType setting is used for basic authentication. SSL data encryption is still used.
  – SSL certificate-based client authentication can be disabled. It means that only the httpAuthType setting (basic authentication) is used.

With these options, SSL can be used for both authentication and data encryption, or just data encryption.

- **Common Manageability Programming Interface support**

  Common Manageability Programming Interface (CMPI) defines a common C-based resource extension interface. Resource extensions can be reused in any management server environment supporting this interface. CMPI is implemented such that the provider can run with any CIM server, not just with Pegasus. The providers use CMPI instead of the Pegasus C/C++ interface. Currently, CMPI supports instance, method, association, and indication providers.

- **Out-of-process provider support**

  Out-of-process (OOP) isolates the providers from the main CIM server by running them in a separate process. All providers are OOP providers for reliability, performance, and security reasons. With OOP, the CIM server does not crash due to a provider crash. Also, the CIM server has granular security control over providers. If one process crashes, it does not cause the crash of other processes and can recover when the next request for that given provider module comes in.

- **DMTF schema 2.14**

  Schema 2.14 contains both experimental and final builds of the schema. This provides you with early access to experimental parts of the model that do not have sufficient implementation experience to be included in the final schema. Experimental elements might change in a backward-incompatible way.

Related information

Common Information Model (CIM) Standards

The Open Group: OpenPegasus

CIM Schema: Version 2.14

Compatibility to V5R3 and V5R4 operating system CIM providers

Classes that are supplied with the IBM Universal Manageability Enablement for i5/OS licensed program are registered into namespace root/cimv2. Some Universal Manageability Enablement CIM providers that are ported from V5R3 and V5R4 operating system CIM providers are also registered into namespace root/ibmsd with a different class name.

These LP-supplied CIM classes that are registered in namespace root/ibmsd have the same parent class and the same properties, and they share the same implementation as classes that are registered in namespace root/cimv2. Only the class name prefixes are different. This makes these CIM providers that are supplied with the IBM Universal Manageability Enablement for i5/OS licensed program compatible to V5R3 and V5R4 operation system CIM providers. In i5/OS V5R3 and V5R4, the operating system CIM classes are registered into namespace root/ibmsd.

The following table shows the class map in namespace root/ibmsd and root/cimv2.

Table 4. Class map in root/ibmsd and root/cimv2

Class names in root/ibmsd	Class names in root/cimv2
IBMPSG_BaseBoard	IBM_BaseBoard
IBMPSG_Chassis	IBM_Chassis
IBMPSG_ComputerSystem	IBMOS400_ComputerSystem
IBMPSG_ComputerSystemDetails	IBM_ComputerSystemDetails
IBMPSG_DirectorAgent	IBM_DirectorAgent
IBMPSG_FRU	IBM_FRU
IBMPSG_Group	IBM_Group
IBMPSG_Lease	IBM_Lease
IBMPSG_NetworkAdapterConfiguration	IBM_NetworkAdapterConfiguration
IBMPSG_NetworkID	IBM_NetworkID
IBMPSG_OperatingSystem	IBMOS400_OperatingSystem
IBMPSG_PhysicalDisk	IBM_PhysicalDisk
IBMPSG_PhysicalMemory	IBM_PhysicalMemory
IBMPSG_PhysicalNetworkAdapter	IBM_PhysicalNetworkAdapter
IBMPSG_Port	IBM_Port
IBMPSG_Processor	IBM_Processor
IBMPSG_SerialNumberInformation	IBM_SerialNumberInformation
IBMPSG_SNMPConfiguration	IBM_SNMPConfiguration
IBMPSG_UserAccount	IBM_UserAccount
IBMPSG_Warranty	IBM_Warranty

Considerations for providers on i5/OS client partitions

You can use the IBM Universal Manageability Enablement for i5/OS licensed program (5722-UME) on i5/OS client partitions, with limitations. Physical hardware information is not accessible to i5/OS partitions that are managed by Integrated Virtualization Manager.

The following tables show the limitations to the functions of the CIM provider.

Table 5. Providers that are inherited from the operating system

CIM classes	Impact
IBMPSG_ComputerSystem	No value returned for properties OtherIdentifyingInfo and IdentifyingDescriptions.
IBMOS400_ComputerSystem	No value returned for properties OtherIdentifyingInfo and IdentifyingDescriptions.
IBM_BaseBoard	No instance returned.
IBM_CSBaseBoard	No instance returned.
IBM_ComputerSystemDetails	No value returned for properties Model and Serial Number.
IBMPSG_Chassis	No instance returned.
IBMPSG_PhysicalDisk	No instance returned.
IBMPSG_PhysicalNetworkAdapter	No instance returned.

Table 5. Providers that are inherited from the operating system (continued)

CIM classes	Impact
IBMPSG_PhysicalMemory	No instance returned.
IBMPSG_Processor	No instance returned.
IBMPSG_FRU	No instance returned.
IBMPSG_Port	No instance returned.
IBMPSG_SerialNumberInformation	No instance returned.

Table 6. Hardware inventory and network management providers, which do not return instances

CIM classes		
IBM_AssociatedMemory	IBM_Card	IBM_CardOnCard
IBM_CDROMDrive	IBM_Chassis	IBM_Chip
IBM_ComputerSystemPackage	IBM_DiskDrive	IBM_DVDDRIVE
IBM_ElementFRU	IBM_Memory	IBM_PackagedComponent
IBM_PackageInChassis	IBM_PCIController	IBM_PCIDevice
IBM_PhysicalMedia	IBM_PhysicalMemory	IBM_PhysicalNetworkAdapter
IBM_Processor	IBM_ProductPhysicalComponent	IBM_Product
IBM_Realizes	IBM_ReplacementFRU	IBM_SystemPackaging

Table 7. SMI-S host bus adapter (HBA) and host-discovered resources (HDR) providers, which do not return instances

CIM classes		
IBM_FCPort	IBM_FCPortStatistics	IBM_Card
IBM_Product	IBM_PortController	IBM_SoftwareIdentity (for firmware)
IBM_SCSIProtocolEndpoint	IBM_FCControlledBy	IBM_FCElementSoftwareIdentity
IBM_FCElementStatisticalData	IBM_FCProductPhysicalComponent	IBM_FCRealizes
IBM_FCSystemDevice	IBM_FCDeviceSAPImplementation	IBM_FCHostedAccessPoint
IBM_SCSIInitiatorTargetLogicalUnitPath		

Providers that are inherited from the operating system

Providers are moved out of the operating system to be ported to i5/OS Portable Application Solutions Environment (i5/OS PASE). This topic describes the implemented CIM class, provider types, and categories for these providers. It also introduces properties, property descriptions, and values of each provider.

Table 8. Providers that are inherited from the operating system

Provider name	Implements CIM class	Provider type	Category
QUME_BootOSFromFSProvider	IBM_BootOSFromFS	Instance and association	OSBase
QUME_ChassisProvider	IBMPSG_Chassis	Instance	Hardware
QUME_ColSrvMetricDefinitionProvider	IBMOS400_ColSrvMetricDefinition IBMOS400_ColSrvMetricDefForME	Instance and association	csMetric

Table 8. Providers that are inherited from the operating system (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_ColSrvMetricValueProvider	IBMOS400_ColSrvMetricValue IBMOS400_ColSrvMetricInstance IBMOS400_ColSrvMetricForME CIM_InstModification	Instance, association, and indication	csMetric
QUME_ComputerSystemDetailsProvider	IBMPSG_ComputerSystemDetails	Instance	Config
QUME_ComputerSystemProvider	IBMOS400_ComputerSystem IBMPSG_ComputerSystem	Instance	OSBase
QUME_CSBaseBoardProvider	IBM_CSBaseBoard	Instance and association	OSBase
QUME_CSNetworkPortProvider	IBM_CSNetworkPort	Instance and association	Network
QUME_CSVirtualProcessorProvider	IBMOS400_CSVirtualProcessor	Instance and association	OSBase
QUME_DirectorAgentProvider	IBMPSG_DirectorAgent	Instance	Software
QUME_DirectorConsumer	/	Consumer	Consumer
QUME_DirectorGroupProvider	IBMPSG_Group	Instance and method	User
QUME_DirectorLeaseEventProvider	IBMPSG_LeaseExpirationEvent	Indication	Event
QUME_DirectorLeaseProvider	IBMPSG_Lease	Instance	Config
QUME_DirectorNetAdaptCfgProvider	IBMPSG_NetworkAdapterConfiguration	Instance and method	Network
QUME_DirectorNetworkIDProvider	IBMPSG_NetworkID	Instance and method	Network
QUME_DirectorUserProvider	IBMPSG_UserAccount	Instance and method	User
QUME_DirectorWarrantyEventProvider	IBMPSG_WarrantyExpirationEvent	Indication	Event
QUME_DirectorWarrantyProvider	IBMPSG_Warranty	Instance	Config
QUME_DiskDriveProvider	IBMPSG_PhysicalDisk	Instance	Hardware
QUME_EthernetPortProvider	IBM_EthernetPort	Instance	Network
QUME_HealthConsumer	IBMPSG_ComponentHealth	Consumer	Consumer
QUME_HostedFileSystemProvider	IBM_HostedFileSystem	Instance and association	OSBase
QUME_IPProtocolEndpointProvider	IBM_IPProtocolEndpoint	Instance	Network
QUME_LocalFileSystemProvider	IBM_LocalFileSystem	Instance	OSBase
QUME_LogConsumer	/	Consumer	Consumer
QUME_NetworkEventProvider	IBMPSG_NetworkAdapterFailedEvent IBMPSG_NetworkAdapterOfflineEvent IBMPSG_NetworkAdapterOnlineEvent	Indication	Event

Table 8. Providers that are inherited from the operating system (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_NetworkFileSystemProvider	IBM_NFS	Instance	OSBase
QUME_NetworkPortProvider	IBMOS400_NetworkPort	Instance	Network
QUME_NWPortImplProtocolEpProvider	IBM_NWPortImplementsIPEndpoint	Instance and association	Network
QUME_OperatingSystemProvider	IBMOS400_OperatingSystem IBMPSG_OperatingSystem	Instance and method	OSBase
QUME_OSProcessProvider	IBMOS400_OSProcess	Instance and association	OSBase
QUME_PhysicalMemoryProvider	IBMPSG_PhysicalMemory	Instance	Hardware
QUME_PhysicalNetworkAdapterProvider	IBMPSG_PhysicalNetworkAdapter	Instance	Hardware
QUME_PhysicalPortProvider	IBMPSG_Port	Instance	Hardware
QUME_ProcessorProvider	IBMPSG_Processor	Instance	Hardware
QUME_ProcessProvider	IBMOS400_Process	Instance	OSBase
QUME_RemoteFileSystemProvider	IBM_RemoteFileSystem	Instance	OSBase
QUME_ReplacementFRUProvider	IBMPSG_FRU	Instance	Hardware
QUME_RunningOSProvider	IBMOS400_RunningOS	Instance and association	OSBase
QUME_SerialNumberProvider	IBMPSG_SerialNumberInformation	Instance	Hardware
QUME_SNMPConfigurationProvider	IBMPSG_SNMPConfiguration	Instance	Config
QUME_SnmpConsumer	/	Consumer	Consumer
QUME_StorageEventProvider	IBMPSG_StorageEvent	Indication	Event
QUME_SystemPackageProvider	IBM_BaseBoard IBMPSG_BaseBoard	Instance	OSBase
QUME_TecConsumer	/	Consumer	Consumer
QUME_TokenRingPortProvider	IBM_TokenRingPort	Instance	Network
QUME_VirtualProcessorProvider	IBMOS400_VirtualProcessor	Instance	OSBase

IBMPSG_ComputerSystem

This provider makes available basic information about the computer system, such as computer name and status information.

Table 9. IBMPSG_ComputerSystem

Property	Property value and data source
string OtherIdentifyingInfo[]	This property returns the following system information: • Type • Serial number • Model • Partition identifier
Name	The system name based on the first entry in the TCP/IP host table.

IBMPSG_BaseBoard

The provider looks up a resource based on the physical resource name that is provided as the key under the Tag property, and returns instances of all backplanes that are available on the system.

Table 10. IBMPSG_BaseBoard

Property name	Property description	Value or value location
boolean HostingBoard	A property that indicates that this card is a system board, or more generically, a baseboard in a chassis.	TRUE
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
boolean Removable	A property that indicates whether a physical package is removable. A physical package is removable if it can be taken in and out of the physical container without impairing the function of the overall packaging.	TRUE
boolean Replaceable	A property that indicates whether a physical package is replaceable. A physical package is replaceable if the element can be replaced with a physically different one.	TRUE
string Caption (64)	A short textual description of the object.	Baseboard *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_BaseBoard
string Description	A class that is derived from the card to deliver the systems baseboard hardware information.	Baseboard information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that is responsible for producing or manufacturing the physical element.	
string Product	The baseboard part number.	Manufacturer IBM
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	
string StatusDescriptions[]	Various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	IBM:*Model*:*SerialNumber*

Table 10. IBMPSG_BaseBoard (continued)

Property name	Property description	Value or value location
uint16 OperationalStatus[]	The current status of the element.	

IBMPSG_Chassis

The provider looks up a resource based on the physical resource name that is provided as the key under the Tag property, and returns instances of all frames that are available on the system.

Table 11. IBMPSG_Chassis

Property name	Property description	Value or value location
boolean AudibleAlarm	A property that indicates whether the frame is equipped with an audible alarm.	FALSE
boolean CanBeFRUed	A property that indicates whether this physical element is a field replaceable unit (TRUE) or not (FALSE).	
boolean IsLocked	A property that indicates whether the frame is currently locked.	FALSE
boolean LockPresent	A property that indicates whether the frame is protected with a lock.	FALSE
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
boolean Removable	A property that indicates whether a physical package is removable. A physical package is removable if it can be taken in and out of the physical container without impairing the function of the overall packaging.	TRUE
boolean Replaceable	A property that indicates whether a physical package is replaceable. A physical package is replaceable if the element can be replaced with a physically different one.	TRUE
boolean VisibleAlarm	A property that indicates whether the equipment includes a visible alarm.	FALSE
string Caption (64)	A short textual description of the object.	Chassis *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_Chassis
string Description	A textual description of the object.	Chassis information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	

Table 11. IBMPSG_Chassis (continued)

Property name	**Property description**	**Value or value location**
string PartNumber (256)	The part number assigned by the organization that is responsible for producing or manufacturing the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	
string StatusDescriptions[]	Various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 PackageType	The type of the physical package.	9 Module or card
uint16 SecurityBreach	An enumerated, integer-valued property that indicates that a physical breach of the frame was attempted but unsuccessful (value=4) or attempted and successful (value=5).	2 (Unknown)

IBMPSG_FRU

The provider looks up a resource based on the physical resource name that is provided as the key under the Name property, and returns instances of physical resources with FRU numbers that are available on the system.

Table 12. IBMPSG_FRU

Property name	**Property description**	**Value or value location**
string Caption (64)	A short textual description of the object.	Field replaceable unit *ElementName*
string Description	A textual description of the object.	Field replaceable unit information for *ElementName*
string ElementName	A user-friendly name for the object.	*Name*
string FRUNumber (key) (64)	FRU ordering information.	
string IdentifyingNumber (key) (64)	FRU identification, such as a serial number on software or a die number on a hardware chip.	
string Name (256)	FRU name.	
string Vendor (key) (256)	The name of the FRU's supplier.	IBM

IBMPSG_PhysicalDisk

The provider looks up a resource based on the logical resource name that is provided as the key under the DeviceID property, and returns instances of logical disk units that are available on the system.

Table 13. IBMPSG_PhysicalDisk

Property name	**Property description**	**Value or value location**
boolean MediaIsLocked	A property that indicates whether the media is locked in the device and cannot be ejected. For devices that cannot be removed, this value should be TRUE.	TRUE
int16 Security	An enumeration that indicates the operational security that is defined for the media access device. For example, information that the device is Read-only (value=4) or information about Boot Bypass (value=6) can be described in this property.	2 (Unknown)
string Caption (64)	A short textual description of the object.	Disk *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_PhysicalDisk
string Description	A textual description of the object.	Disk information for *ElementName*
string DeviceID (key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string ElementName	A user-friendly name of the object	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the element's enabled or disabled state when the EnabledState property is set to 1.	Powered off or not connected
string StatusDescriptions[]	Various OperationalStatus array values.	
string SystemCreationClassName (key) (256)	The scoping system's CreationClassName.	IBMPSG_ComputerSystem
string SystemName (key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault	An enumerated value that indicates an administrator's default configuration for an element's EnabledState.	7 (No Default)
uint16 EnabledState	An integer enumeration that indicates the enabled or disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)

Table 13. IBMPSG_PhysicalDisk (continued)

Property name	Property description	Value or value location
uint64 DefaultBlockSize	The default block size (in bytes) for this device.	
uint64 MaxBlockSize	The maximum block size (in bytes) for media that are accessed by this device.	
uint64 MaxMediaSize	The maximum size (in KB) of media that are supported by this device.	

IBMPSG_PhysicalMemory

The provider looks up a resource based on the physical resource name that is provided as the key under the Tag property, and returns instances of all physical memory resources that are available on the system.

Table 14. IBMPSG_PhysicalMemory

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether this physical element is a field replaceable unit (TRUE) or not (FALSE).	
boolean HasError	A property that indicates whether the memory currently has an error condition.	
boolean IsActive	A property that indicates whether the memory is currently active.	
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
boolean Removable	A property that indicates whether a physical component is removable. A physical component is removable if it can be taken in and out of the physical container without impairing the function of the overall packaging.	TRUE
boolean Replaceable	A property that indicates whether a physical component is replaceable. A physical component is replaceable if the element can be replaced with a physically different one.	TRUE
string Caption (64)	A short textual description of the object.	Physical memory *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_PhysicalMemory
string Description	A textual description of the object.	Physical memory information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	

Table 14. IBMPSG_PhysicalMemory (continued)

Property name	Property description	Value or value location
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that is responsible for producing or manufacturing the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	
string StatusDescriptions[]	Various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 MemoryType	The type of physical memory.	
uint16 OperationalStatus[]	The current status of the element.	
uint32 PositionInRow	The position of the physical memory in a row.	
uint64 Capacity	The total capacity of this physical memory (in bytes).	

IBMPSG_PhysicalNetworkAdapter

The provider looks up a resource based on the physical resource name that is provided as the key under the Tag property, and returns instances of all physical network adapter resources that are available on the system.

Table 15. IBMPSG_PhysicalNetworkAdapter

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether this physical element is a field replaceable unit (TRUE) or not (FALSE).	
boolean HostingBoard	A property that indicates whether this card is a motherboard, or, more generically, a baseboard in a chassis.	FALSE
boolean PoweredOn	A property that indicates whether the physical element is powered on.	

Table 15. IBMPSG_PhysicalNetworkAdapter (continued)

Property name	Property description	Value or value location
boolean Removable	A property that indicates whether a physical package is removable. A physical package is removable if it can be taken in and out of the physical container without impairing the function of the overall packaging.	TRUE
boolean Replaceable	A property that indicates whether a physical package is replaceable. A physical package is replaceable if the element can be replaced with a physically different one.	TRUE
string Caption (64)	A short textual description of the object.	Physical network adapter *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_PhysicalNetworkAdapter
string Description	A textual description of the object.	Physical network adapter information for *ElementName*
string ElementName	A user-friendly name of the objects	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that is responsible for producing or manufacturing the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	

Table 15. IBMPSG_PhysicalNetworkAdapter (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	Various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	

IBMPSG_Port

The provider looks up a resource based on the physical resource name that is provided as the key under the Tag property, and returns instances of all physical ports that are available on the system.

Table 16. IBMPSG_Port

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether this physical element is a field replaceable unit (TRUE) or not (FALSE).	
boolean HotSwappable	A physical component is HotSwappable if it can be replaced by another component within the same model. The component is inserted in a main board that is powered on.	FALSE
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
boolean Removable	A property that indicates whether a physical component is removable. A physical component is removable if it can be taken in and out of the physical container without impairing the function of the overall packaging.	FALSE
boolean Replaceable	A property that indicates whether a physical package is replaceable. A physical component is replaceable if it can be replaced with a physically different one.	FALSE
string Caption (64)	A short textual description of the object	Port connector *ElementName*
String CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_Port

Table 16. IBMPSG_Port (continued)

Property name	Property description	Value or value location
string Description	A textual description of the object.	Port connector information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The Name property that defines the label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that is responsible for producing or manufacturing the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	
String StatusDescriptions[]	Various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 PortType	The type of the port that is presented.	

IBMPSG_Processor

The provider looks up a resource based on the logical resource name that is provided as the key under the DeviceID property, and returns instances of all processors that are available on the system.

Table 17. IBMPSG_Processor

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Processor *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMPSG_Processor
string Description	A textual description of the object.	Processor information for *ElementName*
string DeviceID (key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string ElementName	A user-friendly name of the object.	
string Identifying Descriptions[]	An array of freeform strings that provides explanations and details behind the entries in the OtherIdentifyingInfo array.	The resource name for the logical processor as identified by the Hardware Resource Manager. The processor part number. The processor type number. The processor model number. The processor serial number.

Table 17. IBMPSG_Processor (continued)

Property name	Property description	Value or value location
string Model	The model of the processor.	
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the element's enabled or disabled state when the EnabledState property is set to 1 ("Other").	powered off or not connected
string OtherFamilyDescription	The processor family type.	PowerPC®
string OtherIdentifyingInfo (256)	Additional data, beyond DeviceID information, that can be used to identify a logical device.	
string Role	The role of the processor.	Central Processor
string StatusDescriptions[]	Various OperationalStatus array values.	
string SystemCreationClassName (key) (256)	The scoping system's CreationClassName.	IBMPSG_ComputerSystem
string SystemName (key) (256)	The scoping system's name.	
string Type	The type of the processor.	
string Version	The version of the processor.	
uint16 AddressWidth	The processor address width in bits.	64 bits
uint16 Availability	The primary availability and status of the device.	
uint16 CPUStatus	The current status of the processor.	
uint16 DataWidth	The processor data width in bits.	64 bits
uint16 EnabledDefault	An enumerated value that indicates an administrator's default configuration for an element's enabled state.	7 (No Default)
uint16 EnabledState	An integer enumeration that indicates the enabled or disabled states of an element.	
uint16 Family	The processor family type.	1 (Other)
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element.	Default value of 5 (No change)
uint16 UpgradeMethod	CPU socket information including data on how this processor can be upgraded (if upgrades are supported).	6 (None)

IBMPSG_SerialNumberInformation

This provider returns instances of physical resources of all implemented physical resource classes in this provider that have a serial number associated with them.

Table 18. IBMPSG_SerialNumberInformation

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Serial number for *ElementName*
string Description	A textual description of the object.	Serial number information for *ElementName*
string ElementName	A user-friendly name of the object.	System or *Hardware ElementName*
string Identifier (key)	The identifier by which the asset information object is known.	System or *ResouceName*
string Model	The name of the category by which this element is generally known.	
string Name	The name by which the element that has the given serial number is known.	System or *Hardware ElementName*
string OtherIdentifyingInformation	Additional data, beyond the identifier, that can be used to identify the element.	
string SerialNumber	A manufacturer-allocated number that is used to identify the physical element.	
string SettingId (256)	The identifier by which the setting object is known.	System or *ResouceName*

Hardware inventory and network management providers

These providers provide information about hardware inventory and network management. The tables in the topic indicate which properties are supported and provide property descriptions and values.

Supported providers

The following table lists the implemented CIM class, provider types, and categories for the providers.

Table 19. Supported providers

Provider name	Implements CIM class	Provider type	Category
QUME_AssociatedMemoryProvider	IBM_AssociatedMemory	Instance and association	Logical hardware
QUME_BindsToLANEndpointProvider	IBM_BindsToLANEndpoint	Instance and association	Network
QUME_BIOSElementProvider	IBM_BIOSElement	Instance	Firmware
QUME_CardOnCardProvider	IBM_CardOnCard	Instance and association	Physical hardware
QUME_CardProvider	IBM_Card	Instance	Physical hardware
QUME_CDROMDriveProvider	IBM_CDROMDrive	Instance	Logical hardware
QUME_ChassisProvider	IBM_Chassis	Instance	Physical hardware
QUME_ChipProvider	IBM_Chip	Instance	Physical hardware

Table 19. Supported providers (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_ComputerSystemPackageProvider	IBM_ComputerSystemPackage	Instance and association	Physical hardware
QUME_ControlledByProvider	IBM_ControlledBy	Instance and association	Logical hardware
QUME_DeviceSAPImplementationProvider	IBM_DeviceSAPImplementation	Instance and association	Network
QUME_DiskDriveProvider	IBM_DiskDrive	Instance	Logical hardware
QUME_DNSGeneralSettingDataProvider	IBM_DNSGeneralSettingData	Instance	Network
QUME_DNSSettingDataProvider	IBM_DNSSettingData	Instance	Network
QUME_DVDDriveProvider	IBM_DVDDrive	Instance	Logical hardware
QUME_ElementFRUProvider	IBM_ElementFRU	Instance and association	Physical hardware
QUME_ElementSettingDataProvider	IBM_ElementSettingData	Instance and association	Network
QUME_EthernetPortProvider	IBM_EthernetPort	Instance	Network
QUME_HostedAccessPointProvider	IBM_HostedAccessPoint	Instance and association	Network
QUME_InstalledOSProvider	IBM_InstalledOS	Instance and association	System
QUME_LANEndpointProvider	IBM_LANEndpoint	Instance	Network
QUME_MemoryProvider	IBM_Memory	Instance	Logical hardware
QUME_PackagedComponentProvider	IBM_PackagedComponent	Instance and association	Physical hardware
QUME_PackageInChassisProvider	IBM_PackageInChassis	Instance and association	Physical hardware
QUME_PCIControllerProvider	IBM_PCIController	Instance	Logical hardware
QUME_PCIDeviceProvider	IBM_PCIDevice	Instance	Logical hardware
QUME_PhysicalMediaProvider	IBM_PhysicalMedia	Instance	Physical hardware
QUME_PhysicalMemoryProvider	IBM_PhysicalMemory	Instance	Physical hardware
QUME_PortControllerProvider	IBM_PortController	Instance	Logical hardware
QUME_PortImplementsEndpointProvider	IBM_PortImplementsEndpoint	Instance and association	Network
QUME_ProcessorProvider	IBM_Processor	Instance	Logical hardware
QUME_ProductPhysicalComponentProvider	IBM_ProductPhysicalComponent	Instance and association	Physical hardware
QUME_ProductProvider	IBM_Product	Instance	Physical hardware

Table 19. Supported providers (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_RealizesProvider	IBM_Realizes	Instance and association	Logical hardware
QUME_ReplacementFRUProvider	IBM_ReplacementFRU	Instance	Physical hardware
QUME_SNMPCommunityStringProvider	IBM_SNMPCommunityString	Instance	Network
QUME_SNMPTrapTargetProvider	IBM_SNMPTrapTarget	Instance	Network
QUME_StaticIPAssignmentSettingDataProvider	IBM_StaticIPAssignmentSettingData	Instance	Network
QUME_SystemDeviceProvider	IBM_SystemDevice	Instance and association	Logical hardware
QUME_SystemPackagingProvider	IBM_SystemPackaging	Instance and association	Physical hardware
QUME_TapeDriveProvider	IBMOS400_TapeDrive	Instance	Logical hardware
QUME_TCPProtocolEndpointProvider	IBM_TCPProtocolEndpoint	Instance	Network
QUME_TimeZoneSettingDataProvider	IBM_TimeZoneSettingData	Instance	System
QUME_TokenRingPortProvider	IBM_TokenRingPort	Instance	Network
QUME_WirelessLANEndpointProvider	IBM_WirelessLANEndpoint	Instance	Network
QUME_WirelessPortProvider	IBM_WirelessPort	Instance	Network

IBM_AssociatedMemory

This provider returns the association between a logical element and the memory that is installed on the logical element.

Table 20. IBM_AssociatedMemory

Property name	Property value and data source	Instance mapping rule
IBM_Memory REF Dependent	Returns a reference to the IBM_Memory that is installed or associated with the logical device.	This should be a one-to-*n* association between logical device and memory. It associates each CPU to all main storage. Because it is nonuniform memory access (NUMA) access model, the processor can access any main storage on system.
IBM_Processor REF Antecedent	Returns a reference to the IBM_Processor, representing a logical processor.	

IBM_BindsToLANEndpoint

This provider returns association between a service access point (SAP) or ProtocolEndpoint and an underlying LANEndpoint on the same system.

Table 21. IBM_BindsToLANEndpoint

Property name	Property value and data source	Instance mapping rule
CIM_ServiceAccessPoint REF Dependent	Returns a reference to the CIM_ServiceAccessPoint representing the AccessPoint or ProtocolEndpoint that is dependent on the LANEndpoint property.	This should be a one-to-one association between CIM_IPProtocolEndpoint (which is a subclass of CIM_ServiceAccessPoint) and IBM_LANEndpoint.
IBM_LANEndpoint REF Antecedent	Returns a reference to the IBM_LANEndpoint representing the underlying LANEndpoint property that is depended on.	

IBM_BIOSElement

BIOSElement represents the low-level software that is loaded into nonvolatile storage, and used to start and configure a ComputerSystem. There are three levels of firmware: memory, T-side, and P-side. This provider returns one instance of the active firmware in the memory when an enumerated list of instances is asked for.

Table 22. IBM_BIOSElement

Property name	Property description	Value or value location
string Caption(64)	A short textual description of the object.	BIOS element *ElementName*
string Description	A textual description of the object.	BIOS element information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Manufacturer	CIM_BIOSElement.	IBM
string Name(key)(256)	The name that identifies this software element.	
string SoftwareElementID(256)	An identifier for the software element. The identifier is used with other keys to create a unique representation of the element.	
string Version	Software version. It should be in the form <Major>.<Minor>.<Revision> or <Major>.<Minor><Letter><Revision>.	
uint16 SoftwareElementState(key)(64)	A property that identifies the various states of the life cycle of a software element.	3 (running)
uint16 TargetOperatingSystem(Key)	A property that specifies the element's operating system environment.	i5/OS

IBM_CardOnCard

This provider returns the association between a card and another card or motherboard on which the card is mounted.

Table 23. IBM_CardOnCard

Property name	Property value and data source	Instance mapping rule
IBM_Card REF GroupComponent	Returns a reference to the IBM_Card, representing a card that can hold another card.	This should be a one-to-one association between two cards.
IBM_Card REF PartComponent	Returns a reference to the IBM_Card, representing a card.	
string LocationWithinContainer	Location code.	

IBM_Card

This provider returns instances of all cards that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the ElementName property.

Table 24. IBM_Card

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether a FRU can be applied to this physical element. Its values are TRUE or FALSE.	
boolean HostingBoard	A property that indicates that this card is a motherboard, or more generically, a baseboard in a chassis.	
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
boolean RequiresDaughterBoard	A property that indicates that at least one board or auxiliary card is required to function properly.	
string Caption (64)	A short textual description of the object.	Card *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_Card
string Description	A textual description of the object.	Card information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that produces the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the physical element.	
string SlotLayout	A free-form string that describes the slot positioning, typical usage, restrictions, individual slot spacings, or any other pertinent information for the slots on a card.	

Table 24. IBM_Card (continued)

Property name	**Property description**	**Value or value location**
string StatusDescriptions[]	The various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	Name
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 PackageType	The type of the physical package.	9 Module or Card

IBM_CDROMDrive

This provider returns instances of all CD-ROM drives that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 25. IBM_CDROMDrive

Property name	**Property description**	**Value or value location**
boolean MediaIsLocked	A property that indicates whether the media are locked and cannot be ejected.	TRUE
string Caption (64)	A short textual description of the object.	CDROM *ElementName*
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_CDROMDrive
string Description	A textual description of the object.	CDROM information for *ElementName*
string DeviceID(Key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName(Key) (256)	The scoping System's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7

Table 25. IBM_CDROMDrive (continued)

Property name	Property description	Value or value location
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Security	An enumeration that indicates the operational security defined for the media access device.	2 (Unknown)

IBM_Chassis

Refer to the IBMPSG_Chassis class in the "Providers that are inherited from the operating system" on page 35 topic.

IBM_Chip

This provider returns instances of all chips that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the ElementName property.

Table 26. IBM_Chip

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether a FRU can be applied to this physical element. Its values are TRUE or FALSE.	
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
string Caption (64)	A short textual description of the object.	Chip *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_Chip
string Description	A textual description of the object.	Chip information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that produces the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that is used to identify the Physical Element.	

Table 26. IBM_Chip (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	The various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	

IBM_ComputerSystemPackage

This provider returns the association between a computer system and the physical hardware package that is installed on the system.

Table 27. IBM_ComputerSystemPackage

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalPackage REF Antecedent	Returns a reference to the IBM_PhysicalPackage, representing the physical package that is installed on the system.	This should be a one-to-*n* association between the computer system and the physical package. Enumerate all IBM_PhysicalPackage instances on the system.
IBM_ComputerSystem REF Dependent	Returns a reference to the IBM_ComputerSystem, representing a computer system.	
string PlatformGUID	Physical package's serial number.	

IBM_ControlledBy

This provider returns the association between device and controller.

Table 28. IBM_ControlledBy

Property name	Property value and data source	Instance mapping rule
CIM_Controller REF Antecedent	Returns a reference to the CIM_Controller, representing a controller.	This should be a one-to-one association between a device and a controller.
CIM_LogicalDevice REF Dependent	Returns a reference to the CIM_LogicalDevice, representing a logical port.	
uint16 AccessState		

IBM_DeviceSAPImplementation

This provider returns the association between a service access point (SAP) and how it is implemented.

Table 29. IBM_DeviceSAPImplementation

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Antecedent	Returns a reference to the CIM_LogicalDevice, representing the LogicalDevice.	This should be a one-to-*n* association between CIM_NetworkPort (a subclass of CIM_LogicalDevice) and CIM_IPProtocolEndpoint (a subclass of CIM_ServiceAccessPoint). The QtocLstNetIFc API returns the line description (that maps to an instance of CIM_NetworkPort) for each network interface (that maps to an instance of CIM_IPProtocolEndpoint).
CIM_ServiceAccessPoint REF Dependent	Returns a reference to the CIM_ServiceAccessPoint, representing the ServiceAccessPoint implemented using the LogicalDevice.	

IBM_DiskDrive

This provider returns instances of all logical disk units that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 30. IBM_DiskDrive

Property name	Property description	Value or value location
boolean MediaIsLocked	A property that indicates whether the media are locked in the device and cannot be ejected.	TRUE
string Caption (64)	A short textual description of the object.	Disk Drive *ElementName*
string CreationClassName (Key)(256)	The name of the class or the subclass used in the creation of an instance.	IBM_DiskDrive
string Description	A textual description of the object.	Disk Drive information for *ElementName*
string DeviceID (Key)(64)	An address that uniquely names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key)(256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key)(256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	

Table 30. IBM_DiskDrive (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the Enabled State of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Security	An enumeration that indicates the operational security that is defined for the media access device.	2 (Unknown)
uint64 DefaultBlockSize	The default block size for this device (in bytes).	
uint64 MaxBlockSize	The maximum block size for media that are accessed by this device (in bytes).	
uint64 MaxMediaSize	The maximum size of media that are supported by this device (in KB).	

IBM_DNSGeneralSettingData

This provider returns one instance of this class, which represents the system-wide configuration options for the Domain Name System (DNS) client.

Table 31. IBM_DNSGeneralSettingData

Property name	Property description	Value or value location
string InstanceID(key)	Within the scope of the instantiating namespace, the property that identifies an instance of this class.	IBM_DNSGeneralSettingData
string Caption (64)	A short textual description of the object.	DNSGeneralSettingData
string Description	A textual description of the object.	DNSGeneralSettingData information
string DNSSuffixesToAppend [ ]	A property that appends DNS suffixes to resolve a hostname.	
string ElementName	The user-friendly name for this instance of SettingData.	IBM_DNSGeneralSettingData
uint16 AddressOrigin = 2	A property that identifies the method by which the IP address, subnet mask, and gateway are assigned to the IP protocol endpoint.	2

IBM_DNSSettingData

This provider represents the DNS configuration setting for each TCP/IP interface.

Table 32. IBM_DNSSettingData

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	DNSSettingData
string Description	A textual description of the object.	DNSSettingData information
string DNSServerAddresses [ ]	The DNS servers to be contacted.	
string DomainName	The domain that is used for this client connection.	
string ElementName	The user-friendly name of this instance of SettingData.	IBM_DNSSettingData
string InstanceID(key)	Within the scope of the instantiating namespace, the property that identifies an instance of this class.	IBM_DNSSettingData
string RequestedHostname	The hostname that is requested for this client connection.	
uint16 AddressOrigin = 2	A property that identifies the method by which the IP address, subnet mask, and gateway are assigned to the IP protocol endpoint.	2

IBM_DVDDrive

This provider returns instances of all DVD drives that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 33. IBM_DVDDrive

Property name	Property description	Value or value location
boolean MediaIsLocked	A property that indicates whether the media are locked in the device and cannot be ejected.	TRUE
string Caption (64)	A short textual description of the object.	DVD Drive *ElementName*
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_DVDDrive
string Description	A textual description of the object.	DVD Drive information for *ElementName*
string DeviceID(Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected

Table 33. IBM_DVDDrive (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName(Key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Security	An enumeration that indicates the operational security that is defined for the media access device.	2 (Unknown)

IBM_ElementFRU

This provider returns the association between a physical element and its possible replacement parts.

Table 34. IBM_ElementFRU

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalElement REF ReplaceableElement	Returns a reference to the IBM_PhysicalElement, representing a physical element.	This should be a one-to-*n* association between a physical element and a FRU.
IBM_ReplacementFRU REF ReplacementElement	Returns a reference to the IBM_ReplacementFRU, representing a FRU part.	

IBM_ElementSettingData

This provider returns the association between a LogicalPort and one or more ProtocolEndpoints that are implemented on it.

Table 35. IBM_ElementSettingData

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF ManagedElement	Returns a reference to the CIM_ManagedElement, representing the managed element.	This should be a one-to-one association between CIM_IPProtocolEndpoint (a subclass of CIM_ManagedElement) and IBM_StaticIPAssignmentSettingData (a subclass of CIM_SettingData). For each instance of CIM_IPProtocolEndpoint, enumerate IBM_StaticIPAssignmentSettingData, and if the InstanceID (IPv4 address) of an instance of IBM_StaticIPAssignmentSettingData equals the name of the instance of CIM_IPProtocolEndpoint, use this association to associate these two instances.
CIM_SettingData REF SettingData	Returns a reference to the CIM_SettingData, representing the SettingData object that is associated with the element.	

IBM_EthernetPort

This provider returns instances of all Ethernet line descriptions that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the line description name provided as the key under the DeviceID property.

Table 36. IBM_EthernetPort

Property name	Property description	Value or value location
boolean AutoSense	A property that indicates whether the network port is capable of automatically determining the speed or other communications characteristics of the attached network media.	
boolean FullDuplex	A property that indicates whether the port is operating in full-duplex mode.	
string Caption (64)	A short textual description of the object.	Ethernet port *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_EthernetPort
string Description	A textual description of the object.	Ethernet port information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string NetworkAddresses [ ] (64)	An array of strings that indicates the network addresses for the port.	
string PermanentAddress (64)	The network address that is hardcoded into a port.	
string StatusDescriptions[]	The various OperationalStatus array values.	

Table 36. IBM_EthernetPort (continued)

Property name	Property description	Value or value location
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key)(256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 LinkTechnology	An enumeration of the types of links.	2 Ethernet
uint16 OperationalStatus[]	The current status of the element.	
uint16 PortNumber	The network port number. Network ports are often numbered relative to either a logical module or a network element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	
uint32 MaxDataSize	The maximum size of the INFO (non-MAC) field that is received or transmitted.	
uint64 ActiveMaximumTransmissionUnit	The active or negotiated maximum transmission unit (MTU) that can be supported.	
uint64 MaxSpeed	The maximum bandwidth of the port in bits per second.	
uint64 RequestedSpeed	The requested bandwidth of the port in bits per second.	*Speed*
uint64 Speed	The bandwidth of the port in bits per second.	
uint64 SupportedMaximumTransmissionUnit	The maximum transmission unit (MTU) that can be supported.	

IBM_HostedAccessPoint

This provider returns the association between a service AccessPoint and the system on which it is provided.

Table 37. IBM_HostedAccessPoint

Property name	Property value and data source	Instance mapping rule
CIM_ServiceAccessPoint REF Dependent	Returns a reference to the CIM_ServiceAccessPoint, representing the SAPs that are hosted on this system.	This should be a one-to-*n* association between IBM_ComputerSystem and CIM_ServiceAccessPoint. Enumerate all CIM_ServiceAccessPoint properties on the system.
IBM_ComputerSystem REF Antecedent	Returns a reference to the IBM_ComputerSystem, representing the hosting system.	

IBM_InstalledOS

This provider returns the association between the ComputerSystem and the OperatingSystem operating systems installed or loaded on it.

Table 38. IBM_InstalledOS

Property name	Property value and data source	Instance mapping rule
IBM_ComputerSystem REF GroupComponent	Returns a reference to the IBM_ComputerSystem, representing the ComputerSystem.	This should be a one-to-*n* association between IBM_ComputerSystem and IBM_OperatingSystem. Enumerate all IBM_OperatingSystem properties on the system
IBM_OperatingSystem REF PartComponent	Returns a reference to the IBM_OperatingSystem, representing the OperatingSystem installed on the ComputerSystem.	

IBM_LANEndpoint

This provider returns instances of all line descriptions on the system when an enumerated list of instances is asked for, or looks up a resource based on the line description name provided as the key under the ElementName property

Table 39. IBM_LANEndpoint

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	LANEndpoint *ElementName*
string CreationClassName(Key)(256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_LANEndpoint
string Description	A textual description of the object.	LANEndpoint information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string GroupAddresses [ ]	The multicast addresses to which the LANEndpoint listens.	
string MACAddress(12)	The principal unicast address that is used in communication with the LANEndpoint.	
string Name(256)	A string that identifies this protocol endpoint with either a port or an interface on a device.	
string NameFormat (256)	The name that ensures that the value of the name property is unique.	LineName_MACAddress

Table 39. IBM_LANEndpoint (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key)(256)	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string SystemName(Key)(256)	The name of the scoping System.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 ProtocolIFType	Enumeration is limited to layer 2 values that are related and reserved for this subclass of protocol endpoint.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	
uint32 MaxDataSize	The largest information field that can be sent or received by the LANEndpoint.	

IBM_Memory

This provider returns instances of all memory-related logical devices available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 40. IBM_Memory

Property name	Property description	Value or value location
boolean Volatile	A property that indicates whether this memory is volatile.	TRUE
string Caption (64)	A short textual description of the object.	Cache Memory *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_Memory
string Description	A textual description of the object.	Cache Memory information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	*ElementName*
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected

Table 40. IBM_Memory (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
uint16 Access	A property that indicates whether the media are readable, writable, or both.	**1** Read **2** Write **3** Read and write
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the enabled state of an element.	7 (No default)
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)

IBM_PackagedComponent

This provider returns the association between a physical component and a physical package that the component belongs to.

Table 41. IBM_PackagedComponent

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalPackage REF GroupComponent	Returns all instances of CIM_PhysicalPackage.	This should be a one-to-one association between a component and a package.
CIM_PhysicalComponent REF PartComponent	Returns all instances of CIM_PhysicalComponent.	
string LocationWithinContainer	Location code.	

IBM_PackageInChassis

This provider returns the association between a physical package and a chassis that the package belongs to.

Table 42. IBM_PackageInChassis

Property name	Property value and data source	Instance mapping rule
IBM_Chassis REF GroupComponent	Returns all instances of IBM_Chassis.	This should be a one-to-one association between a package and a chassis.
CIM_PhysicalPackage REF PartComponent	Returns all instances of CIM_PhysicalPackage.	
string LocationWithinContainer	Location code.	

IBM_PCIController

This provider returns instances of all PCI controllers that are available on the system when an enumerated list of instances is asked for, or looks up the resource based on the logical resource name provided as the key under the DeviceID property.

Table 43. IBM_PCIController

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	PCI Controller *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_PCIController
string Description	A textual description of the object.	PCI Controller information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5

IBM_PCIDevice

This provider returns instances of all PCI Devices available on the system when an enumerated list of instances is asked for, or looks up a resource based on the Logical Resource Name provided as the key under the DeviceID property.

Table 44. IBM_PCIDevice

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	PCI Device *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_PCIDevice
string Description	A textual description of the object.	PCI Device information for *ElementName*
string DeviceID (Key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5

IBM_PhysicalMedia

This provider returns instances of all physical media that is available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the ElementName property.

Table 45. IBM_PhysicalMedia

Property name	Property description	Value or value location
boolean CanBeFRUed	A property that indicates whether a FRU can be applied to this physical element. Its values are TRUE and FALSE.	
boolean PoweredOn	A property that indicates whether the physical element is powered on.	
real32 MediaSize	The size of the media in inches.	
string Caption (64)	A short textual description of the object.	Physical Media *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_PhysicalMedia
string Description	A textual description of the object.	Physical Media information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Model (256)	The name by which the physical element is generally known.	
string Name (1024)	The label by which the object is known.	
string PartNumber (256)	The part number assigned by the organization that produces the physical element.	
string SerialNumber (256)	A manufacturer-allocated number that identifies the physical element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that uniquely identifies the physical element and serves as the element's key.	*Name*
uint16 HealthState	The current health of the element.	
uint16 MediaType	The type of the physical media as an enumerated integer.	
uint16 OperationalStatus[]	The current status of the element.	
uint64 Capacity	The number of bytes that can be read from or written to the medium.	

IBM_PhysicalMemory

This provider returns instances of all physical memory that is available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the ElementName property.

Refer to the IBMPSG_PhysicalMemory class in the "Providers that are inherited from the operating system" on page 35 topic.

IBM_PortController

This provider returns instances of all port controllers that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 46. IBM_PortController

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Port Controller *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_PortController
string Description	A textual description of the object.	Port Controller information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 ControllerType	The type or model of the port controller.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the Enabled State of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5

IBM_PortImplementsEndpoint

This provider returns the association between a LogicalPort and one or more ProtocolEndpoints that are implemented on it

Table 47. IBM_PortImplementsEndpoint

Property name	Property value and data source	Instance mapping rule
CIM_LogicalPort REF Antecedent	Returns a reference to the CIM_LogicalPort, representing the device behind the ProtocolEndpoint.	This should be a one-to-*n* association between CIM_NetworkPort (a subclass of CIM_LogicalPort) and CIM_IPProtocolEndpoint (a subclass of CIM_ProtocolEndpoint).
CIM_ProtocolEndpoint REF Dependent	Returns a reference to the CIM_ProtocolEndpoint, representing the ProtocolEndpoint that is implemented on the LogicalPort.	

IBM_Processor

Refer to the IBMPSG_Processor class in the Providers that are inherited from the base operating system topic.

IBM_ProductPhysicalComponent

This provider returns the association between a physical element and the product that it belongs to.

Table 48. IBM_ProductPhysicalComponent

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalElement REF PartComponent	Returns a reference to the IBM_PhysicalElement, representing a physical element.	This should be a one-to-one association between physical element and a product. Enumerate all CIM_PhysicalElement instances and map to IBM_Product.
IBM_Product REF GroupComponent	Returns a reference to the IBM_Product, representing a product that contains the physical element.	

IBM_Product

This provider returns instances of all products available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the ElementName property.

Table 49. IBM_Product

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Product *Name*
string Description	A textual description of the object.	Product information for *Name*
string ElementName	A user-friendly name of the object.	
string IdentifyingNumber (key)(64)	Product identification, such as the serial number on software, the die number on a hardware chip, or a project number.	
string Name (Key)(256)	Commonly used product name.	
string Vendor (Key)(256)	The name of the product’s supplier.	””
string Version (Key)(64)	Product version information.	

IBM_Realizes

This provider returns the association between logical devices and physical elements that implement them.

Table 50. IBM_Realizes

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Dependent	Returns all instances of CIM_LogicalDevice.	This should be a one-to-*n* association between CIM_LogicalDevice and CIM_PhysicalElement.
CIM_PhysicalElement REF Antecedent	Returns all instances of CIM_PhysicalElement.	

IBM_ReplacementFRU

This provider returns instances of all replacement FRUs available on the system when an enumerated list of instances is asked for, or looks up a resource based on the packaging resource name provided as the key under the InstanceID property.

Table 51. IBM_ReplacementFRU

Property name	Property description	Value or value location
boolean CustomerReplaceable	Indicates whether this replacement part is considered customer replaceable (TRUE) or not (FALSE).	
string Caption (64)	A short textual description of the object.	Field replaceable unit *ElementName*
string Description	A textual description of the object.	Field replaceable unit information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string InstanceID (Key)	Within the scope of the instantiating Namespace, the property that identifies an instance of this class.	*ElementName*

IBM_SNMPCommunityString

This provider contains the controlling information of accessing SNMP Service.

Table 52. IBM_SNMPCommunityString

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	SNMP Community String
string CommunityString;	The SNMP community string or password that is used for read access or read-and-write access to the agent's data.	
string CreationClassName	The name of the class or the subclass that is used in the creation of an instance.	IBM_SNMPCommunityString
string Description	A textual description of the object.	SNMP Community string information for *SystemName*
string ElementName	A user-friendly name of the object.	*TrapTargetIP*
string SystemCreationClassName	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem

Table 52. IBM_SNMPCommunityString (continued)

Property name	Property description	Value or value location
string SystemName	The name of the scoping system.	
uint16 TypeOfAccess	An enumerated integer that describes whether read access or read-and-write access is granted, or whether this information is unknown.	
string Name	The uniquely identified ServiceAccessPoint	*TrapTargetIP*

IBM_SNMPTrapTarget

This provider contains information that describes a remote system to which Informs and Traps are sent.

Table 53. IBM_SNMPTrapTarget

Property name	Property description	Value or value location
String AccessInfo	The host address.	
string Caption (64)	A short textual description of the object.	SNMP Trap Target
string CommunityString;	The SNMP community string or password that is used for read access, or read-and-write access to the agent's data.	
string CreationClassName	The name of the class or the subclass that is used in the creation of an instance.	IBM_SNMPTrapTarget
string Description	A textual description of the object.	SNMP Trap Target information for *SystemName*
string ElementName	A user-friendly name of the object.	*AccessInfo*
string Name	A property that identifies the service access point.	*AccessInfo*
string SystemCreationClassName	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string SystemName	The name of the scoping system.	
uint16 SNMPVersion	A property that indicates whether read access or read-and-write access is granted, or whether this information is unknown.	

IBM_StaticIPAssignmentSettingData

This provider returns instances of all static IP AssignmentSettingData available on the system when an enumerated list of instances is asked for, or looks up a resource based on the IPv4 address provided as the key under the InstanceID property.

Table 54. IBM_StaticIPAssignmentSettingData

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Static IP Assignment Setting Data *ElementName*

Table 54. IBM_StaticIPAssignmentSettingData (continued)

Property name	Property description	Value or value location
string Description	A textual description of the object.	Static IP Assignment Setting Data information for *ElementName*
string ElementName	The user-friendly name of this instance of SettingData.	*IPv4Address*
string GatewayIPv4Address[]	The IPv4 addresses of the default gateway.	
string InstanceID(key)	Within the scope of the instantiating namespace, the property that identifies an instance of this class.	*IPv4Address*
string IPv4Address	The IPv4 address that is assigned to the protocol endpoint.	
string SubnetMask	The subnet mask for the IPv4 address of this protocol endpoint.	
uint16 AddressOrigin = 3	The method by which the IP address, subnet mask, and gateway are assigned to the IP protocol endpoint.	3

IBM_SystemDevice

This provider returns the association between a ComputerSystem and all LogicalDevice instances on it.

Table 55. IBM_SystemDevice

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF PartComponent	Returns all instances of CIM_LogicalDevice.	This should be a one-to-*n* association between IBM_ComputerSystem and CIM_LogicalDevice. Enumerate all CIM_LogicalDevice properties on the computer system.
IBM_ComputerSystem REF GroupComponent	Returns an instance of IBM_ComputerSystem.	

IBM_SystemPackaging

This provider returns the association between a computer system and all physical packages on it.

Table 56. IBM_SystemPackaging

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalElement REF Antecedent	Returns all instances of CIM_PhysicalElement	This should be a one-to-*n* association between IBM_ComputerSystem and CIM_LogicalDevice. Enumerate all CIM_PhysicalElement properties on the computer system.
IBM_ComputerSystem REF Dependent	Returns an instance of IBM_ComputerSystem.	

IBM_TCPProtocolEndpoint

This provider returns instances of all TCP connections on the system when an enumerated list of instances is asked for, or looks up a resource based on the local address and port number provided as the key under the name property.

Table 57. IBM_TCPProtocolEndpoint

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	TCP Protocol Endpoint *ElementName*
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_TCPProtocolEndpoint
string Description	A textual description of the object.	TCP Protocol Endpoint information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Name(256)	A string that identifies this protocol endpoint with either a port or an interface on a device.	
string NameFormat (256)	The name that ensures that the value of the name property is unique.	Local Address_Local Port
string SystemCreationClassName (Key)(256)	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string SystemName(Key)(256)	The name of the scoping system.	
uint16 ProtocolIFType	Enumeration is limited to TCP and reserved values for this subclass of ProtocolEndpoint.	4111
uint32 PortNumber	The TCP port number.	

IBM_TimeZoneSettingData

This provider returns instances of all time zone settings when an enumerated list of instances is asked for, or looks up a resource based on the time zone description name provided as the key under the InstanceID property.

Table 58. IBM_TimeZoneSettingData

Property name	Property description	Value or value location
sint32 DaylightOffset	The number of minutes by which this daylight saving time differs from UTC.	
sint32 StandardOffset	The number of minutes by which this standard time differs from UTC.	
string Caption (64)	A short textual description of the object.	Time Zone Setting Data for *ElementName*
string DaylightName	The full name of the daylight time zone.	
string Description	A textual description of the object.	Time Zone Setting Data information for *ElementName*
string ElementName	A user-friendly name of the object.	
string InstanceID(key)	Within the scope of the instantiating namespace, this property identifies an instance of this class.	*ElementName*
string StandardName	The full name of the standard time zone.	

IBM_TokenRingPort

This provider returns instances of all TokenRing line descriptions that are available on the system when an enumerated list of instances is asked for, or looks up a resource based on the line description name provided as the key under the DeviceID property.

Table 59. IBM_TokenRingPort

Property name	Property description	Value or value location
boolean AutoSense	A property that indicates whether the network port is capable of automatically determining the speed or other communications characteristics of the attached network media.	
boolean FullDuplex	A property that indicates whether the port is operating in full-duplex mode.	
string Caption (64)	A short textual description of the object.	TokenRing port *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_TokenRingPort
string Description	A textual description of the object.	TokenRing port information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string NetworkAddresses [ ] (64)	An array of strings that indicates the network addresses for the port.	
string PermanentAddress (64)	The network address that is hardcoded into a port.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 LinkTechnology	An enumeration of the types of links.	7 Token Ring
uint16 OperationalStatus[]	The current status of the element.	

Table 59. IBM_TokenRingPort (continued)

Property name	Property description	Value or value location
uint16 PortNumber	The network port number. Network ports are often numbered relative to either a logical module or a network element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	
uint16 RingSpeed	The bandwidth of the ring.	
uint32 MaxDataSize	The maximum size of the INFO (non-MAC) field that is received or transmitted.	
uint64 ActiveMaximumTransmissionUnit	The active or negotiated MTU that can be supported.	
uint64 MaxSpeed	The maximum bandwidth of the port in bits per second.	
uint64 RequestedSpeed	The requested bandwidth of the port in bits per second.	*Speed*
uint64 Speed	The bandwidth of the port in bits per second.	
uint64 SupportedMaximumTransmissionUnit	The MTU that can be supported.	

IBM_WirelessLANEndpoint

This provider returns instances of all wireless line descriptions on the system when an enumerated list of instances is asked for, or looks up a resource based on the line description name provided as the key under the Elementname property

Table 60. IBM_WirelessLANEndpoint

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Wireless LAN Endpoint *ElementName*
string CreationClassName(Key)(256)	The name of the class or the subclass used in the creation of an instance.	IBM_WirelessLANEndpoint
string Description	A textual description of the object.	Wireless LAN Endpoint information for *ElementName*
string ElementName	A user-friendly name of the object.	*Name*
string Name	The MAC address of the wireless endpoint.	
string NameFormat (256)	The name that ensures that the value of the name property is unique.	LineName_MACAddress
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key)(256)	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string SystemName(Key)(256)	The name of the scoping system.	

Table 60. IBM_WirelessLANEndpoint (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 ProtocolIFType	Enumeration is limited to layer 2 values that are related and reserved for this subclass of ProtocolEndpoint.	71
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	

IBM_WirelessPort

This provider returns instances of all wireless ports available on the system when an enumerated list of instances is asked for, or looks up a resource based on the logical resource name provided as the key under the DeviceID property.

Table 61. IBM_WirelessPort

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Wireless port *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_WirelessPort
string Description	A textual description of the object.	Wireless port information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string NetworkAddresses [ ] (64)	An array of strings that indicates the network addresses for the port.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string PermanentAddress (64)	The network address that is hardcoded into a port.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key)(256)	The scoping system's name.	

Table 61. IBM_WirelessPort (continued)

Property name	Property description	Value or value location
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 LinkTechnology	An enumeration of the types of links.	Wireless LAN 11
uint16 OperationalStatus[]	The current status of the element.	
uint16 PortNumber	Network port number Network ports are often numbered relative to either a logical module or a network element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5

IBMOS400_TapeDrive

This provider returns instances of all tape drives available on the system when an enumerated list of instances is asked for, or looks up a resource based on the Logical Resource Name provided as the key under the DeviceID property.

Table 62. IBMOS400_TapeDrive

Property name	Property description	Value or value location
real32 MediaSize	The size of media in inches.	
string Caption (64)	A short textual description of the object.	Tape *ElementName*
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMOS400_TapeDrive
string Description	A textual description of the object.	Tape information for *ElementName*
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string ElementName	A user-friendly name of the object.	*Name*
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	The enabled or disabled state of the element when the EnabledState property is set to 1.	powered off or not connected
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	

Table 62. IBMOS400_TapeDrive (continued)

Property name	Property description	Value or value location
uint16 Availability	The primary availability and status of the device.	
uint16 EnabledDefault = 2	An enumerated value that indicates an administrator's default or startup configuration for the EnabledState of an element.	7
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 HealthState	The current health of the element.	
uint16 MediaType	An enumerated integer that specifies the type of physical media.	
uint16 OperationalStatus[]	The current status of the element.	
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Security	An enumeration that indicates the operational security defined for the media access device.	2 (unknown)

Supported CIM SMI-S HBA and HDR providers

In the IBM Universal Manageability Enablement for i5/OS program, the Common Information Model Object Manager (CIMOM) uses profiles host bus adapter (HBA) and host-discovered resources (HDR). This topic describes the IBM-supplied Storage Management Initiative Specification (SMI-S) HBA and HDR providers.

SMI-S is a standard management interface that allows interoperability of different storage area network (SAN) resources that are provided by different vendors. SMI-S is based on the Common Information Model (CIM) and Web-Based Enterprise Management (WBEM) standards that are originated from the Distributed Management Task Force (DMTF). The SMI-S Server Profile is mandatory for all compliant SMI-S servers.

SAN is a dedicated network that is separated from a local area network (LAN) and a wide area network (WAN). SAN generally refers to interconnected storage-related resources that are connected to one or more servers. It is characterized by high-data transmission rates between the computer system and member storage elements.

Figure 1 through 5 outline elements and their association for HBA, HDR, and registered profiles.

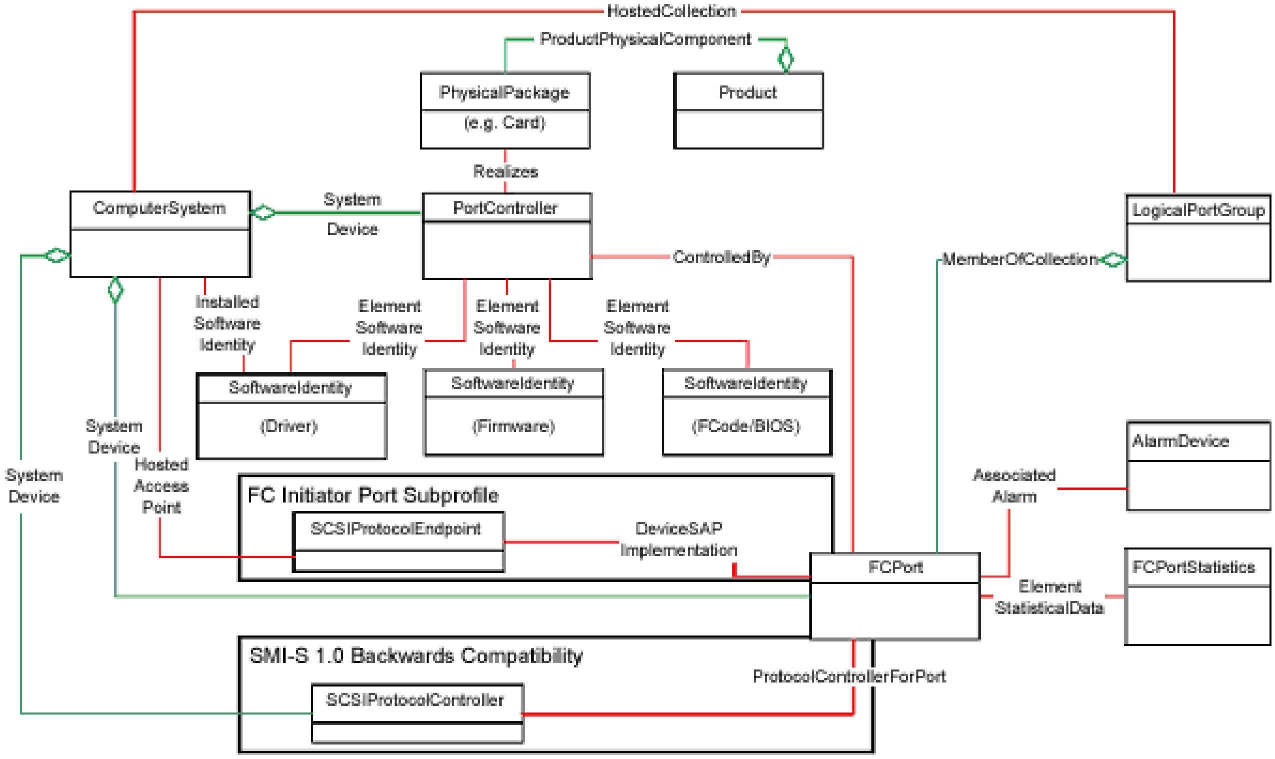

Figure 1. HBA profile

The HBA profile describes behavior of Fibre Channel (FC) host adapters supporting the SCSI (FC SCSI Protocol (FCP)) command set. An FC adapter that is used in a host system is called an HBA. An HBA is a physical device that contains one or more FC ports. A single system contains one or more HBAs.

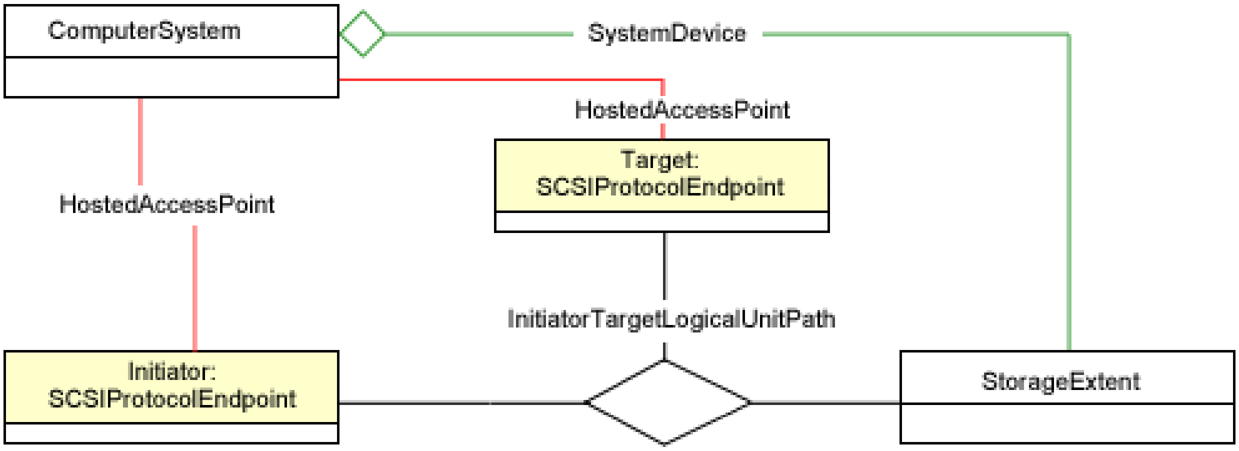

Figure 2. HDR profile

The HDR profile provides information about the discovered hardware resources that include the connectivity and corresponding IDs.

HDR profiles are extended to model i5/OS auxiliary storage pools (ASPs), mirroring, and multiple paths. Figure 3 and figure 4 show the modelling of i5/OS ASPs, mirroring, and multiple paths.

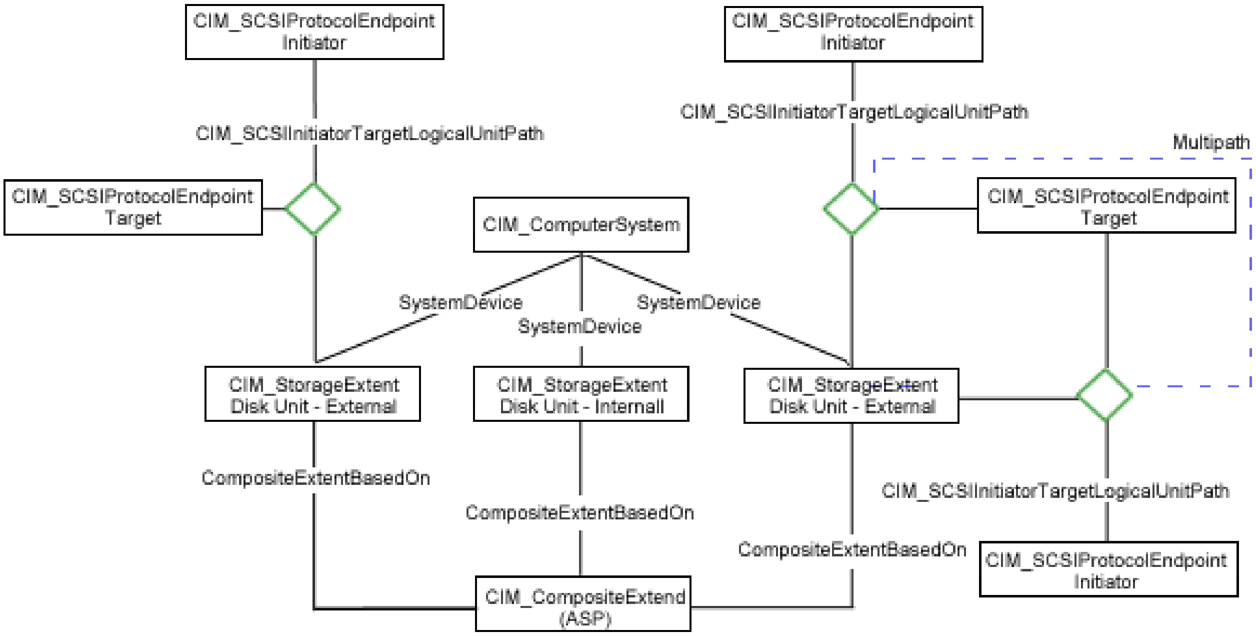
CIM_SCSIProtocolEndpoint Initiator
CIM_SCSIProtocolEndpoint Initiator
CIM_SCSIInitiatorTargetLogicalUnitPath
CIM_SCSIInitiatorTargetLogicalUnitPath
Multipath
CIM_SCSIProtocolEndpoint Target
CIM_ComputerSystem
CIM_SCSIProtocolEndpoint Target
SystemDevice
SystemDevice
SystemDevice
CIM_StorageExtent Disk Unit - External
CIM_StorageExtent Disk Unit - Internall
CIM_StorageExtent Disk Unit - External
CompositeExtentBasedOn
CompositeExtentBasedOn
CIM_SCSIInitiatorTargetLogicalUnitPath
CompositeExtentBasedOn
CIM_SCSIProtocolEndpoint Initiator
CIM_CompositeExtend (ASP)

Figure 3. CIM representation of a nonmirrored ASP

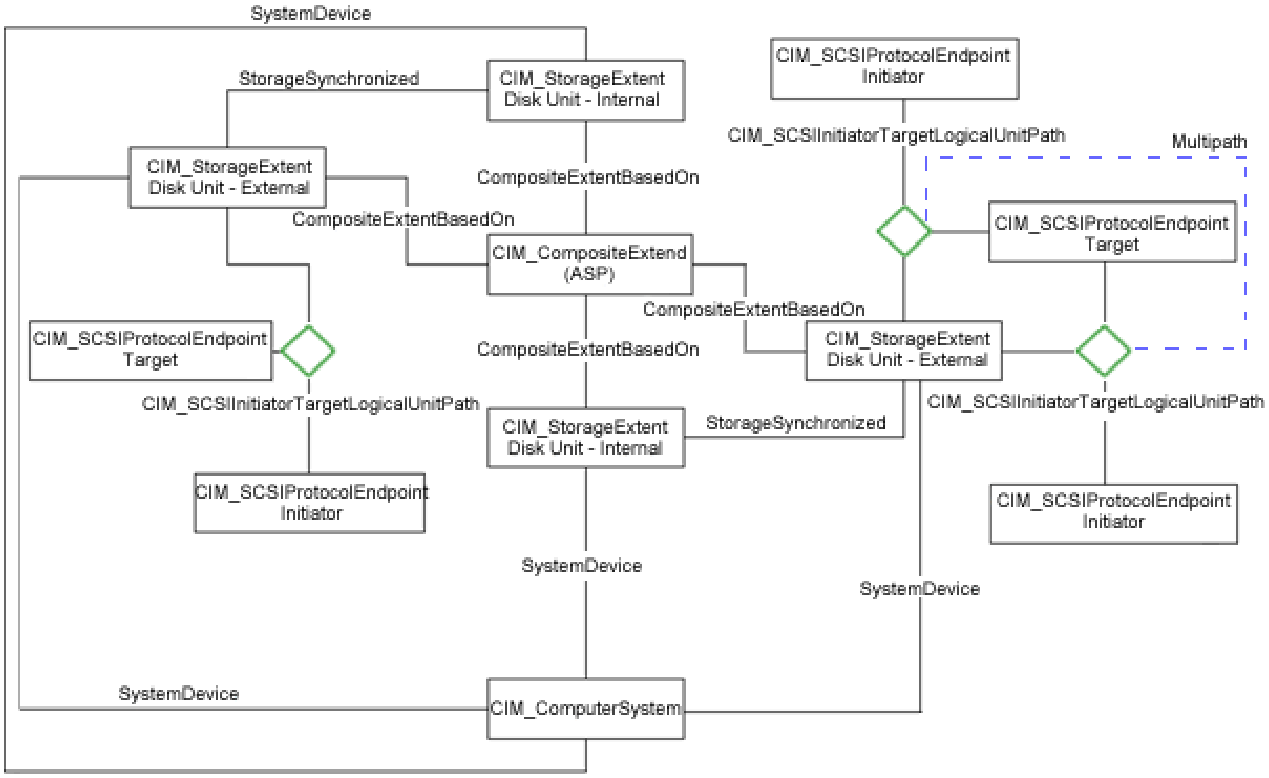
SystemDevice
StorageSynchronized
CIM_StorageExtent Disk Unit - Internal
CIM_SCSIProtocolEndpoint Initiator
CIM_SCSIInitiatorTargetLogicalUnitPath
Multipath
CIM_StorageExtent Disk Unit - External
CompositeExtentBasedOn
CompositeExtentBasedOn
CIM_CompositeExtend (ASP)
CIM_SCSIProtocolEndpoint Target
CompositeExtentBasedOn
CIM_SCSIProtocolEndpoint Target
CompositeExtentBasedOn
CIM_StorageExtent Disk Unit - External
CIM_SCSIInitiatorTargetLogicalUnitPath
CIM_SCSIInitiatorTargetLogicalUnitPath
CIM_StorageExtent Disk Unit - Internal
StorageSynchronized
CIM_SCSIProtocolEndpoint Initiator
CIM_SCSIProtocolEndpoint Initiator
SystemDevice
SystemDevice
SystemDevice
CIM_ComputerSystem

Figure 4. CIM representation of a mirrored ASP

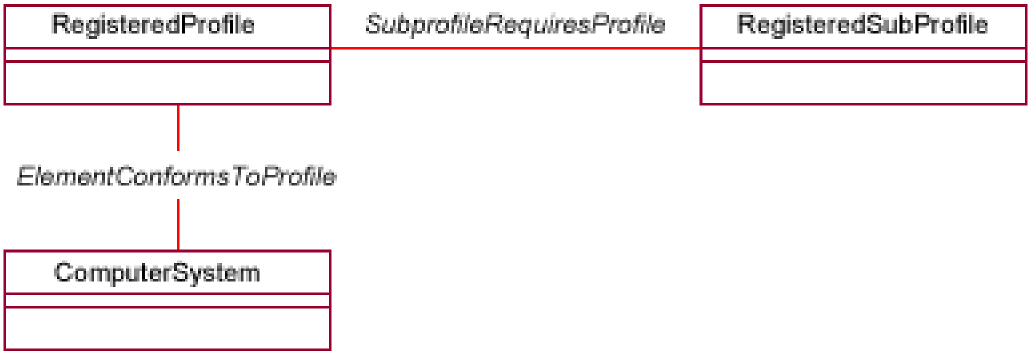

Figure 5. Registered profile

Registered profile models the profiles that are registered in the object manager and the associations between registration classes and the domain classes implementing the profile.

Note: The green lines and the red lines in the preceding figures represent two types of associations:

- The green lines represent the aggregation association.
- The red lines represent the non-aggregation association.

CIM instance providers

The following CIM classes have been implemented as IBM-supplied providers to discover HBA card and storage resources on the host:

- IBM_Card: subclass of CIM_Card
- IBM_CompositeExtent: subclass of CIM_CompositeExtent
- IBM_ComputerSystem: subclass of CIM_Computer_System
- IBM_FCPort: subclass of CIM_FCPort
- IBM_FCPortStatistics: subclass of CIM_ FCPortStatistics
- IBM_PortController: subclass of CIM_PortController
- IBM_Product: subclass of CIM_Product
- IBM_RegisteredProfile: subclass of CIM_ RegisteredProfile
- IBM_RegisteredSubProfile: subclass of CIM_RegisteredSubProfile
- IBM_SoftwareIdentity: subclass of CIM_SoftwareIdentity
- IBM_SCSIProtocolEndPoint: subclass of CIM_SCSIProtocolEndPoint
- IBM_StorageExtent: subclass of CIM_StorageExtent

CIM association providers

The following CIM classes have been implemented as IBM-supplied providers to provide association information for discovered HBA card and storage resources on the host.

- IBM_ASPSystemDevice: subclass of CIM_SystemDevice
- IBM_CompositeExtentBasedOn: subclass of CIM_CompositeExtentBasedOn
- IBM_ElementConformsToProfile: subclass of CIM_ElementConformsToProfile
- IBM_FCControlledBy: subclass of CIM_ControlledBy
- IBM_FCDeviceSAPImplementation (Optional): subclass of CIM_DeviceSAPImplementation
- IBM_FCElementSoftwareIdentity: subclass of CIM_ElementSoftwareIdentity
- IBM_FCElementStatisticalData: subclass of CIM_ElementStatisticalData
- IBM_FCHostedAccessPoint : subclass of CIM_HostedAccessPoint

- IBM_FCProductPhysicalComponent: subclass of CIM_ProductPhysicalComponent
- IBM_FCRealizes: subclass of CIM_Realizes
- IBM_FCSystemDevice: subclass of CIM_SystemDevice
- IBM_SCSIInitiatorTargetLogicalUnitPath: subclass of CIM_SCSIInitiatorTargetLogicalUnitPath
- IBM_StorageSynchronized: subclass of CIM_StorageSynchronized
- IBM_SubProfileRequiresProfile: subclass of CIM_SubProfileRequiresProfile

SMI-S HBA CIM providers

IBM_Card

The IBM_Card provider returns the HBA physical card instance on the system.

Table 63. IBM_Card

Property name	Property description	Value or value location
boolean PoweredOn	This property indicates whether the physical element is powered on (TRUE) or is powered off (FALSE).	
boolean RequiresDaughterBoard	This property indicates that at least one daughter board or auxiliary card is required to function properly.	
string Caption (64)	A short textual description of the object.	FC Card *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass used in the creation of an instance.	IBM_Card
string Description	A textual description of the object.	FC Card Information *ElementName*
string ElementName	A user-friendly name of the object.	Resource name
string Manufacturer (256)	The name of the organization that produces the physical element.	IBM
string Model (256)	The name by which the physical element is generally known.	Type-Model
string Name (1024)	The label by which the object is known.	Resource name
string SerialNumber (256)	A manufacturer-allocated number that identifies the physical element.	
String SlotLayout	A freeform string that describes slot positioning, typical usage, restrictions, individual slot spacings, or any other pertinent information for the slots on a card.	
String StatusDescriptions	The various OperationalStatus array values.	
string Tag (key) (256)	An arbitrary string that identifies the physical element and serves as the key of the element.	Manufacturer_Type-Model-SerialNumber
uint16 OperationalStatus	The current status of the element.	
uint16 PackageType	An enumeration that defines the type of the physical package.	9 Module or Card

IBM_ComputerSystem

Refer to the IBM_ComputerSystem provider in the “Providers that are inherited from the operating system” on page 35 topic for more information.

IBM_FCPort

IBM_FCPort provider returns instances of all the FC ports on the system.

Note: The HBA profile does not cover the internal storage controller and virtual HBA.

Table 64. IBM_FCPort

Property name	Property description	Value or value location
string Description	A textual description of the object.	FC Port information *DeviceID*
string Caption (64)	A short textual description of the object.	IBM FCPort
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_FCPort
string DeviceID (key) (64)	An address that names the logical device.	
string Name (1024)	The label by which the object is known.	
string PermanentAddress (64)	The network address that is hardcoded into a port.	
string SystemCreationClassName (key) (256)	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName (key) (256)	The system name of the scoping system.	
uint16 LinkTechnology	An enumeration of the types of links.	4 FC
uint16 OperationalStatus	The current status of the element.	
uint16 PortType	The specific mode that is currently enabled for the port.	10 N
uint16 SupportedFC4Types	An array of integers that indicates the Fibre Channel (FC)-4 protocols supported.	8 SCSI-FCP
uint16 UsageRestriction	A logical port is identifiable as a front-end or back-end port.	3 Back-end only
uint16[] ActiveFC4Types	A textual description of the object.	8 SCSI - FCP
uint64 MaxSpeed	The maximum bandwidth of the port in bits per second.	0
uint64 Speed	The current bandwidth of the port in bits per second.	0

IBM_FCControlledBy

The IBM_ControlledBy provider returns the association between a device and a controller.

Table 65. IBM_FCControlledBy

Property name	Property value and data source	Instance mapping rule
IBM_FCPort REF Dependent	The provider returns a reference to the CIM_LogicalDevice. This reference represents a logical port.	This should be a one-to-*n* association between the controller and FCPort.
IBM_PortController REF Antecedent	The provider returns a reference to the CIM_Controller. This reference represents a controller.	

IBM_FCDeviceSAPImplementation

This provider is an association between a service access point (SAP) and how it is implemented.

Table 66. IBM_FCDeviceSAPImplementation

Property name	Property value and data source	Instance mapping rule
IBM_FCPort REF Antecedent	The logical device.	This should be a one-to-one association between IBM_FCPort and ″initiator″ IBM_SCSIProtocolEndpoint.
IBM_SCSIProtocolEndpoint REF Dependent	The SAP that is implemented using the logical device.	

IBM_FCElementSoftwareIdentity

This provider allows a managed element to report its software-related asset information (firmware, drivers, configuration software, and so forth).

Table 67. IBM_FCElementSoftwareIdentity

Property name	Property value and data source	Instance mapping rule
IBM_PortController REF Dependent	The managed element that requires or uses the software.	This should be a one-to-*n* association between the controller and SoftwareIdentity.
IBM_SoftwareIdentity REF Antecedent	The software asset of a logical element.	

IBM_FCElementStatisticalData

This provider is an association that relates a managed element to its statistical data.

Table 68. IBM_FCElementStatisticalData

Property name	Property value and data source	Instance mapping rule
IBM_FCPortREF ManagedElement	The managed element for which statistical or metric data is defined.	This should be a one-to-one association between FCPort and FCPortStatistics.
IBM_FCPortStatistics REF Stats	The statistic information (object).	

IBM_FCPortStatistics

This provider returns statistical data of all Fibre Channel (FC) ports on the system.

Table 69. IBM_FCPortStatistics

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	IBM HBA Port Statistics
string Description	A textual description of the object.	This class represents instances of the statistics for HBA Ports
string ElementName	A user-friendly name of the object.	
string InstanceID (key)	Within the scope of the instantiating namespace, InstanceID identifies an instance of this class.	
uint64 BytesReceived	The total number of bytes that are received, including framing characters.	0
uint64 BytesTransmitted	The total number of bytes that are transmitted, including framing characters.	0
uint64 CRCErrors	The number of times that the cyclic redundancy check (CRC) in a frame does not match the CRC that is computed by the receiver.	0
uint64 InvalidTransmissionWords	The number of transmission words with characters that are not valid.	0
uint64 LinkFailures	The number of times that a link error has occurred.	0
uint64 LossOfSignalCounter	The number of times that the signal is lost on the port since the last reset of the device.	0
uint64 LossOfSyncCounter	The number of times that synchronization is lost on the port since the last reset of the device.	0
uint64 PacketsReceived	The total number of packets that are received.	0
uint64 PacketsTransmitted	The total number of packets that are transmitted.	0
uint64 PrimitiveSeqProtocolErrCount	The count of primitive sequence protocol errors that are detected at this port.	0

IBM_FCProductPhysicalComponent

This provider returns an association between a physical element and the product that it belongs to.

Table 70. IBM_FCProductPhysicalComponent

Property name	Property value and data source	Instance mapping rule
IBM_Card REF PartComponent	The provider returns a reference to the IBM_PhysicalElement. The reference represents a physical element.	This should be a one-to-one association between physical element and the product. Enumerate all CIM_PhysicalElement instances and map to IBM_Product.
IBM_Product REF GroupComponent	The provider returns a reference to the IBM_Product. The reference represents a product that contains the physical element.	

IBM_FCRealizes

This provider returns an association between logical devices and physical elements that implement them.

Table 71. IBM_FCRealizes

Property name	Property value and data source	Instance mapping rule
IBM_Card REF Antecedent	The provider returns all instances of CIM_PhysicalElement.	This should be a one-to-*n* association between CIM_LogicalDevice and CIM_PhysicalElement.
IBM_PortController REF Dependent	The provider returns all instances of CIM_LogicalDevice.	

IBM_FCSystemDevice

This provider returns an association between a computer system and all logical devices on it.

Table 72. IBM_FCSystemDevice

Property name	Property value and data source	Instance mapping rule
IBM_ComputerSystem REF GroupComponent	The provider returns an instance of CIM_System.	This should be a one-to-*n* association between IBM_ComputerSystem and IBM_FCPort.
IBM_FCPort REF PartComponent	The provider returns all instances of CIM_LogicalDevice.	

IBM_FCHostedAccessPoint

This provider is an association between an SAP and the system on which it is provided.

Table 73. IBM_FCHostedAccessPoint

Property name	Property value and data source	Instance mapping rule
IBM_ComputerSystem REF Antecedent	The hosting system.	This should be a one-to-*n* association between IBM_ComputerSystem and initiator SCSIProtocolEndpoint.
IBM_SCSIProtocolEndpoint REF Dependent	The SAPs that are hosted on this system.	

IBM_PortController

This provider returns instances of all port controllers available on the system when an enumerated list of instances is asked for, or it looks up a resource based on the logical resource name that is provided as the key under the DeviceID property.

Table 74. IBM_PortController

Property name	Header	Header
string Caption (64)	A short textual description of the object.	Port Controller *ElementName*
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_PortController
String Description	A textual description of the object.	Port Controller information for *ElementName*
string DeviceID (key) (64)	An address that names the logical device.	
string ElementName	A user-friendly name of the object.	Resource name
string Name (1024)	The label by which the object is known.	
string OtherEnabledState	This property describes the enabled or disabled state of the element when the EnabledState property is set to 1.	not connected if EnabledState is 1; "" if EnabledState is not 1.
string StatusDescriptions	The various OperationalStatus array values.	
string SystemCreationClassName (key) (256)	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName (key) (256)	The system name of the scoping system.	HostName of the System
uint16 ControllerType	The type or model of the port controller.	4 FC
uint16 EnabledDefault	An enumerated value that indicates an administrator's default or startup configuration for the Enabled State of an element.	7 (No Default)
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 OperationalStatus	The current status of the element.	
uint16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element.	5 (no change)

IBM_Product

This provider returns instances of all products available on the system when an enumerated list of instances is asked for, or it looks up the resource based on the packaging resource name that is provided as the key under the ElementName property.

Table 75. IBM_Product

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Product *ElementName*
string Description	A textual description of the object.	Product information for *ElementName*
string ElementName	A user-friendly name of the object.	Vendor_Type-Model
string IdentifyingNumber (key) (64)	Product identification, such as a serial number on software, a die number on a hardware chip, or a project number.	Serial Number
string Name (key) (256)	Commonly used product name.	Storage IOA
string Vendor (key) (256)	The name of the product supplier.	IBM
string Version (key) (64)	Product version information.	Type-Model

IBM_SCSIProtocolEndPoint

This provider returns the Small Computer System Interface (SCSI) protocol supported by the HBA card on the system.

Table 76. IBM_SCSIProtocolEndPoint

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	IBM FC SCSI Protocol EndPoint
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_SCSIProtocolEndpoint
string Description	A textual description of the object.	This class represents instances of available SCSI protocol over an FC port.
string Name (256)	The label by which the object is known.	
string OtherTypeDescription (64)	The type of ProtocolEndpoint when the Type property of this class (or any of its subclasses) is set to 1.	
string SystemCreationClassName (key) (256)	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName (key) (256)	The system name of the scoping system.	HostName of the System
uint16 ConnectionType	The supported connection type for this endpoint.	2 Fibre Channel
uint16 ProtocolIFType	An enumeration that is synchronized with the IANA ifType MIB.	56 Fibre Channel
uint16 Role	For iSCSI, each SCSI protocol endpoint must act as either a target or an initiator endpoint.	2 Initiator

IBM_SoftwareIdentity

This provider returns the device driver and device firmware instance on the system.

Table 77. IBM_SoftwareIdentity firmware

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	IBM FC Adapter Software Identity
string Description	A textual description of the object.	This class represents instances of available fibre channel adapter software entities.
string InstanceID (key)	Within the scope of the instantiating namespace, InstanceID identifies an instance of this class.	SLIC_5722999
string Manufacturer	The manufacturer of this software.	IBM
string VersionString	A string that represents the complete software version information.	SLIC VRM
uint16 []Classifications	An array of enumerated integers that classifies this software.	2 Driver

Table 78. IBM_SoftwareIdentity driver

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	IBM FC Adapter Software Identity
string Description	A textual description of the object.	This class represents instances of available fibre channel adapter software entities.
string InstanceID (key)	Within the scope of the instantiating namespace, InstanceID opaquely and uniquely identifies an instance of this class.	
string Manufacturer	The manufacturer of this software.	IBM
string VersionString	A string that represents the complete software version information.	LID
uint16 []Classifications	An array of enumerated integers that classifies this software.	10 Firmware

SMI-S HDR CIM classes

IBM_ASPSystemDevice

This provider represents the association between a computer system and the auxiliary storage pool (ASP).

Table 79. IBM_ASPSystemDevice

Property name	Property value and data source	Instance mapping rule
IBM_CompositeExtent REF PartComponent	This property returns all instances of CIM_LogicalDevice.	This should be a one-to-*n* association between CIM_System and CIM_LogicalDevice. Enumerate all CIM_LogicalDevice on the system
IBM_ComputerSystem REF GroupComponent	This property returns an instance of CIM_System.	

IBM_CompositeExtent

This provider is used to model the distribution of user data across one or more underlying StorageExtents, which might be protected by some redundancy mechanism.

Table 80. IBM_CompositeExtent

Property name	Property description	Value or value location
boolean IsBasedOnUnderlyingRedundancy	This property indicates that the underlying StorageExtents participate in a StorageRedundancy group.	Geographical mirroring
boolean NoSinglePointOfFailure	This property indicates whether any single point of failure exists.	Mirrored level of protection
string Caption	A short textual description of the object.	ASP *ElementName*
string CreationClassName	The name of the class or the subclass that is used in the creation of an instance.	IBM_CompositeExtent
string Description	A textual description of the object.	ASP information for *ElementName*
string DeviceID	An address that names the logical device.	ASP *ASP number*
string ElementName	A user-friendly name for the object.	ASP name
string Name(Experimental)	A unique identifier for the extent.	ASP *ASP number*
string SystemCreationClassName	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName	The system name of the scoping system.	HostName of the System
string[] IdentifyingDescriptions	An array of freeform strings that provides explanations and details behind the entries in the OtherIdentifyingInfo array.	Auxiliary Storage Pool
string[] OtherIdentifyingInfo	This property captures data.	ASP
uint8 DeltaReservation	The current value for delta reservation.	Tracking Space/StorageCapacity × 100
uint16 DataRedundancy	The number of complete copies of data that is currently maintained.	Geographical Mirroring (either 1 or 2)
uint16 NameFormat(Experimental)	This property requires that logical disk names must use the operating system device name format.	12
uint16 NameNamespace(Experimental)	This property requires that logical disk names must use the operating system device namespace.	8

Table 80. IBM_CompositeExtent (continued)

Property name	Property description	Value or value location
uint16 PackageRedundancy	This property indicates how many physical packages can currently fail without data loss.	0
uint16[] OperationalStatus	The current status of the element.	ASP - OK, IASP - OK (Varied On), Stopped (Varied Off)
uint64 BlockSize	The size (in bytes) of the blocks that form this StorageExtent.	Block Size of a StorageExtent in the ASP
uint64 ConsumableBlocks	The maximum number of blocks that are available for consumption when layering StorageExtents using the BasedOn association.	Capacity - Must be converted from Number of Pages
uint64 NumberOfBlocks	The total number of logically contiguous blocks that form this extent.	Capacity - Must be converted from Number of Pages

IBM_CompositeExtentBasedOn

This class indicates that two storage objects are replicated at the specified point in time.

Table 81. IBM_CompositeExtentBasedOn

Property name	Property value and data source	Instance mapping rule
CIM_CompositeExtent REF Dependent	The CompositeExtent that is built on the StorageExtent.	
CIM_StorageExtent REF Antecedent	The underlying StorageExtent.	

IBM_SCSIInitiatorTargetLogicalUnitPath

This provider is an association that models a host driver path to a SCSI logical unit. Each permutation of initiator, target protocol endpoints, and logical units is considered a separate path. This provider describes end-to-end path behavior, such as properties and operations that are commonly used in multipath management.

Table 82. IBM_SCSIInitiatorTargetLogicalUnitPath

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF LogicalUnit	A subclass of a logical device that represents a SCSI logical unit.	
IBM_SCSIProtocolEndpoint REF Initiator	An initiator endpoint.	
IBM_SCSIProtocolEndpoint REF Target	A target endpoint.	

IBM_SCSIProtocolEndPoint

This provider returns the SCSI protocol endpoint for the disk unit on the system.

Table 83. IBM_SCSIProtocolEndPoint

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	IBM FC SCSI Protocol EndPoint
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_SCSIProtocolEndpoint
string Description	A textual description of the object.	This class represents instances of available SCSI protocol over FC port
string Name (256)	The label by which the object is known.	World wide unique logical unit identifier
string OtherTypeDescription (64)	The type of protocol endpoint when the Type property of this class (or any of its subclasses) is set to 1.	
string SystemCreationClassName (key) (256)	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName (key) (256)	The system name of the scoping system.	HostName of the System
uint16 ConnectionType	The supported connection type for this endpoint.	2 Fibre Channel
uint16 ProtocolIFType	An enumeration that is synchronized with the IANA ifType MIB.	56 Fibre Channel
uint16 Role	For iSCSI, each SCSIProtocolEndpoint must act as either a target or an initiator endpoint.	3 Target

IBM_StorageExtent

This provider describes the capabilities and management of the various media that exist to store data and allow data retrieval.

Table 84. IBM_StorageExtent

Property name	Property description	Value or value location
Boolean IsBasedOnUnderlyingRedundancy	If the value is set to true, the property indicates that the underlying StorageExtents participate in a StorageRedundancy group.	Mirroring and RAID are true; otherwise, this is false.
boolean NoSinglePointOfFailure	This property indicates whether any single point of failure exists.	Mirrored level of protection is true if the level of protection is bus.
string Description	A textual description of the object.	
string Caption (64)	A short textual description of the object.	IBM Storage Extent
string CreationClassName (key) (256)	The name of the class or the subclass used in the creation of an instance.	IBM_StorageExtent
string DeviceID (key) (64)	An address that names the logical device.	Serial Number

Table 84. IBM_StorageExtent (continued)

Property name	Property description	Value or value location
string ElementName	A user-friendly name of the object.	Resource name, or concatenating all the resource names of multipathed disk units, separated by commas.
string Name(Experimental)	A unique identifier of the extent.	Resource Name
string SystemCreationClassName (key) (256)	The CreationClassName of the scoping system.	IBM_ComputerSystem
string SystemName (key) (256)	The system name of the scoping system.	HostName of the System
string[] IdentifyingDescriptions	An array of freeform strings that provides explanations and details behind the entries in the OtherIdentifyingInfo array.	Resource Name
string[] OtherIdentifyingInfo (256)	This property captures data besides the device ID information that can be used to identify a logical device.	
uint8 DeltaReservation	The current value for delta reservation.	0
uint16 DataRedundancy	The number of complete copies of data that is currently maintained.	Mirroring equals 2, otherwise 1
uint16 ExtentStatus	This property indicates that StorageExtents have additional status information beyond what is captured in the OperationalStatus and other properties that are inherited from ManagedSystemElement.	
uint16 NameFormat(Experimental)	This property requires that logical disk names must use the operating system device name format.	12 (OS Device Name)
uint16 NameNamespace(Experimental)	This property requires that logical disk names must use the operating system device namespace.	8 (OS Device Namespace)
uint16 PackageRedundancy	This property indicates how many physical packages can currently fail without data loss.	Mirroring equals 1, RAID 5 equals 1, RAID 6 equals 2.
uint16[] OperationalStatus	The current statuses of the element.	
uint64 BlockSize	The size (in bytes) of the blocks that form this StorageExtent.	Block Size
uint64 ConsumableBlocks	The maximum number of blocks that are available for consumption when layering StorageExtents using the BasedOn association.	Size in Sectors
uint64 NumberOfBlocks	The total number of logically contiguous blocks that form this Extent.	Size in Sectors

IBM_StorageSynchronized

This provider indicates that two storage objects were replicated at the specified point in time.

Table 85. IBM_StorageSynchronized

Property name	Property value and data source	Instance mapping rule
IBM_StorageExtent REF SystemElement	The source of the replication.	The mirrored disk unit has the same unit number in i5/OS.
IBM_StorageExtent REF SystemElement	The target of the replication.	

SMI-S Registered Profile CIM classes

IBM_ElementConformsToProfile

This association defines the RegisteredProfiles to which the referenced ManagedElement conforms.

Table 86. IBM_ElementConformsToProfile

Property name	Property value and data source	Instance mapping rule
IBM_ComputerSystem REF ManagedElement	The ManagedElement that conforms to the RegisteredProfile.	RegisteredProfile FC HBA associates to ComputerSystem; RegisteredProfile HDR associates to ComputerSystem.
CIM_RegisteredProfile REF ConformantStandard	The RegisteredProfile to which the ManagedElement conforms.	

IBM_RegisteredProfile

This provider returns instances of all RegisteredProfiles of HBA and HDR.

Table 87. IBM_RegisteredProfile

Property name	Property description	Value or value location
string Caption	A short textual description (one-line string) of the object.	*RegisteredName*
string Description	Provides a textual description of the object.	Registered Profile information *RegisteredName*
string ElementName	A user-friendly name for the object.	*RegisteredName*
string InstanceID	Within the scope of the instantiating namespace, the InstanceID that identifies an instance of this class.	IBMOS400 *RegisteredName*
string RegisteredName	The name of this registered profile.	FC HBA or HDR
string RegisteredVersion	The version of this profile.	1.1.0
uint16 AdvertiseTypes [ ]	This property signifies the advertisement for the profile information.	3 SLP
uint16 RegisteredOrganization	The organization that defines this profile.	11 SNIA

IBM_RegisteredSubProfile

This provider returns instances of all SubProfiles that are supported by HBA and HDR profiles.

Table 88. IBM_RegisteredSubProfile

Property name	Property description	Value or value location
string Caption	A short textual description (one-line string) of the object.	*RegisteredName*
string Description	A textual description of the object.	Registered Sub Profile information *RegisteredName*
string ElementName	A user-friendly name of the object.	*RegisteredName*
string InstanceID	Within the scope of the instantiating namespace, the InstanceID that identifies an instance of this class.	IBMOS400 *RegisteredName*
string RegisteredName	The name of this registered profile.	FC Initiator Ports Subprofile
string RegisteredVersion	The version of this profile.	1.1.0
uint16 AdvertiseTypes [ ]	This property signifies the advertisement for the profile information.	3 SLP
uint16 RegisteredOrganization	The organization that defines this profile.	11 SNIA

IBM_SubProfileRequiresProfile

This provider is an association between the RegisteredProfile and its subprofiles.

Table 89. IBM_SubProfileRequiresProfile

Property name	Property value and data source	Instance mapping rule
IBM_RegisteredProfile REF Antecedent	The RegisteredProfile that is referenced or required by the subprofile.	RegisteredProfile FC HBA supports FC Initiator Ports Subprofile.
IBM_RegisteredSubProfile REF Dependent	A RegisteredSubProfile that requires a scoping profile for context.	

i5/OS metrics classes

The topic describes metric classes and the user authorization.

The following CIM classes have been implemented as IBM-supplied providers to provide performance information:

- IBMOS400_ColSrvMetricDefinition: a subclass of CIM_BaseMetricDefinition
- IBMOS400_ColSrvMetricValue: a subclass of CIM_BaseMetricValue
- IBMOS400_ColSrvMetricInstance: a subclass of CIM_MetricInstance that associates between metric definition and metric value.
- IBMOS400_ColSrvMetricDefForME: a subclass of CIM_MetricDefForME that associates between a managed element (resource) and metric definition.
- IBMOS400_ColSrvMetricForME: a subclass of CIM_MetricForME that associates between a managed element (resource) and metric value.

Note: All instances of IBMOS400_ColSrvMetricValue return volatile data, and only current data is supported. Historical data is not supported.

The following figure illustrates CIM standard schemas classes, metric providers association resources classes, and i5/OS extended metric classes.

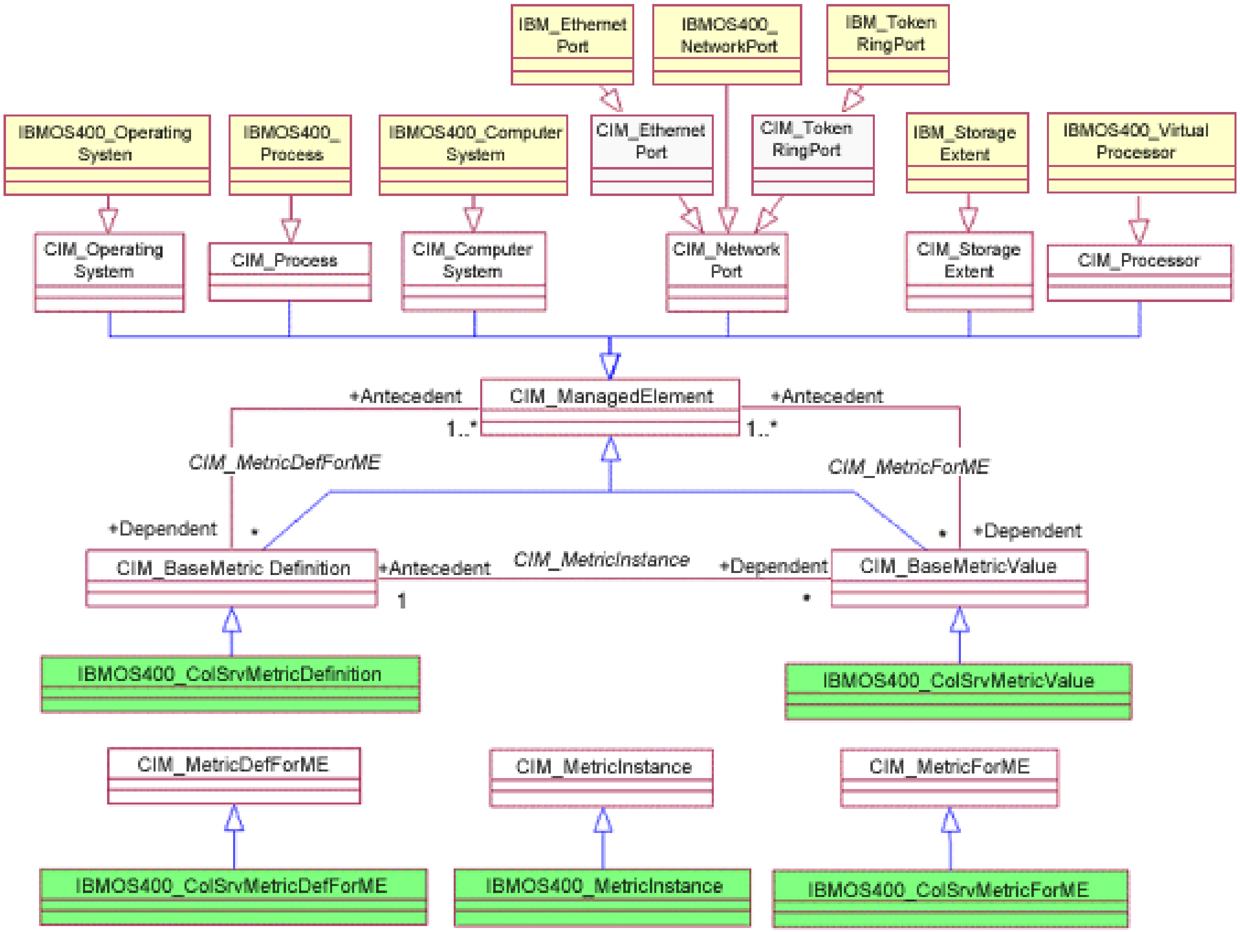

Figure 6. CIM metrics classes

Notes:

- The white class diagrams are CIM standard schemas classes.
- The yellow class diagrams are metric providers association resources class.
- The green class diagrams are i5/OS extended metric classes.

The operating system supports the following CIM metrics.

Table 90. CIM metrics

Resource/Base CIM class	Metric
IBM_StorageExtent	**ActiveTimePercentage** Percentage the disk unit was active processing some request, id:MDDS0E **AvailableSpace** Free capacity on the disk unit, id:MDDS05 **AverageDeviceUtilization** Average device utilization (not normalized to 100% for parallel I/O activity), id:MDDS0D **Capacity** Capacity of the disk unit, id:MDDS04 **FastWriteOperations** Number of fast write operations (stored first in nonvolatile memory), id:MDDS08 **FastWritePercentage** Percentage of write operations processed as fast write operations, id:MDDS09 **IOIntensity** I/O utilization indicator, id:MDDS10 **QueueDepth** Average number of I/O requests currently in queue (OS view), id: MDDS11 **ReadCacheHitPercentage** Percentage of read requests that did not need access to disk units, id:MDDS03 **ReadOperations** Number of read operations against the disk unit, id:MDDS06 **ReadThroughput** Bytes per second read, id:MDDS0A **RequestRate** Number of I/O requests per second for the associated device, id:MDDS0F **ResponseTime** ResponseTime associated with a disk unit, id:MDDS01 **TransferredThroughput** Bytes per second transferred, id:MDDS0C **WaitTime** WaitTime associated to with disk unit, id:MDDS02 **WriteOperations** Number of write operations, id:MDDS07 **WriteThroughput** Bytes per second written, id:MDDS0B

Table 90. CIM metrics (continued)

Resource/Base CIM class	Metric
IBMOS400_ComputerSystem	**ActiveVirtualProcessors** Average number of virtual processors active, id: MDCS21 **PctPartitionDefinedCapacityUsed** System CPU time used as a percentage of configured capacity (the amount of CPU the logical partition is configured to use), id: MDCS01 **UnusedGlobalCPUCapacity** CPU time in milliseconds not used on global server level, id: MDCS23 **UnusedPartitionCPUCapacity** Reserved but unused capacity for this operating system container, id: MDCS22
IBMOS400_NetworkPort , IBM_EthernetPort, IBM_TokenRingPort	**BytesReceived** The total number of bytes received, including framing characters, id: MDNP02 **BytesTransmitted** The total number of bytes transmitted, including framing characters, id: MDNP01 **ErrorRate** Number of network errors per second, id: MDNP03 **Percentage of capacity actually used** Id: MDNP04

Table 90. CIM metrics (continued)

Resource/Base CIM class	Metric
IBMOS400_OperatingSystem	**CPUConsumptionIndex** CPU time used divided by CPU time that might have been used by this operating system, id: MDOS25 **ExternalViewKernelModePercentage** i5/OS always returns 0, id: MDOS2C **ExternalViewTotalCPUPercentage** External view CPU percentage, id: MDOS2A **ExternalViewUserModePercentage** External view user mode percentage, id: MDOS2B **FreePhysicalMemory** i5/OS always returns 0, id: MDOS04 **FreeSpaceInPagingFiles** Free space in system ASP, id: MDOS05 **FreeVirtualMemory** Free space in system auxiliary storage pool (ASP), id: MDOS03 **InternalViewIdlePercentage** Idle percentage as seen from within the operating system, id: MDOS24 **InternalViewKernelModePercentage** i5/OS always returns 0, id: MDOS21 **InternalViewTotalCPUPercentage** User mode percentage as seen from within the operating system, id: MDOS23 **InternalViewUserModePercentage** i5/OS always returns 0, id: MDOS22 **KernelModeTime** i5/OS always returns 0, id: MDOS09 **NumberOfProcesses** Number of jobs active during the sample interval, id: MDOS02 **NumberOfUsers** Number of interactive jobs active during the sample interval, id: MDOS01 **OperationalStatus** i5/OS always returns `OK`, id: MDOS08 **PageInRate** Number of pages that are paged in per second in all pools, id: MDOS06 **TotalCPUTime** Same as UserModeTime, id: MDOS0B **UserModeTime** Total system CPU time used, id: MDOS0A

Table 90. CIM metrics (continued)

Resource/Base CIM class	Metric
IBMOS400_Process	**AccumulatedKernelModeTime** i5/OS always returns 0, id: MDPR29 **AccumulatedTotalCPUTime** CPU time spent for this process since process creation, id: MDPR2B **AccumulatedUserModeTime** CPU time in user mode spent for this process since process creation, id: MDPR2A **ExternalViewTotalCPUPercentage** External view total CPU percentage, id: MDPR26 **ExternalViewKernelModePercentage** i5/OS always returns 0, id: MDPR28 **ExternalViewUserModePercentage** External view user mode percentage, id: MDPR27 **InternalViewKernelModePercentage** i5/OS always returns 0, id: MDPR23 **InternalViewTotalCPUPercentage** Percentage value related to TotalCPUTime, id: MDPR25 **InternalViewUserModePercentage** Percentage value related to UserModeTime, the percentage the system CPUs were used for this process in user mode during the measurement interval, id: MDPR24 **KernelModeTime** i5/OS always returns 0, id: MDPR01 **TotalCPUTime** Same as UserModeTime, id: MDPR03 **UserModeTime** The CPU time used by the JOB (including all secondary threads), id: MDPR02
IBMOS400_VirtualProcessor	**TotalCPUTimePercentage** The time a virtual processor was used as a percentage of the elapsed interval time, id: MDPC01

Metrics definition instance provider

The Metrics definition data is no longer stored in the CIM repository, but an explicit instance provider capability is added to the metrics definition provider.

id This property of a metric definition class must be generated and be unique for every instance of metric definition.

Note: The id value is added in the preceding table.

Metrics authorization

Metrics providers need to run with QSECOFR authority so that collection services and the performance database work properly. For the nonrequestor providers, Application Administration is used instead of authorizing users to each object that they access. The advantage of Application Administration is that authorization failures are automatically audited by the system.

The user function registration APIs manage the registration and usage information for functions. To help manage systems, the user function registration APIs provide a mechanism for registering functions and controlling which users are allowed to use those functions. The control of user functions, however, is not a replacement for securing resources. Users who are not allowed to use a particular function are not prevented from accessing a resource through another interface. See User Function Registration APIs for more information.

A metrics value provider (including its instance and association capabilities) uses application administration to authorize users to every CIM operation using single registration entry. When operations are routed to the metric value providers, application administration determinates whether the user is authorized to that operation. If not authorized, an access denied exception is thrown and a message is returned to tell the user why access is denied. If authorized, the user is permitted to access the metric values.

Note: If a user disables the authentication of the CIM server (set enableAuthentication as false), the metrics authorization is disabled automatically at the same time. It means that all users can use the capabilities of metrics value providers.

The metrics definition provider has lower security expectations for users, so all its instances and associations can be accessed by any user without checking authorization.

Related concepts

"User authorization on CIMOM" on page 17
User authorization is a type of security check that verifies whether you have access to the objects you want to change. Authorization is needed not only for changing operations but sometimes for reading operations as well.

Related tasks

"Configuring CIMOM" on page 4
When you install the i5/OS operating system, the IBM Universal Manageability Enablement for i5/OS licensed program is installed on the system by default. Before you use the CIM function of this licensed program, you need to configure the Common Information Model Object Manager (CIMOM).

Related information

User Function Registration APIs

i5/OS support for the CIM indication provider

You can use the CIM metric indication provider to notify applications when a specific metric event occurs.

The CIM indication provider notifies user applications when specified metric data occurs on the server that the provider supervises. Each application must subscribe to the provider by providing, in query form, information about an event about which it wants data. Examples of events are occurrences such as authentication failures, disk-write errors, or even mouse clicks. The provider then notifies the application when the event occurs. Such an occurrence is called an *indication*. When metrics match client-submitted queries, the indication provider creates the indication and returns it to the client.

Important: The metric indication provider only accepts queries that filter on either the InstanceId or the MetricDefinitionId property. The provider rejects empty filters or a filter that provides only properties other than these two.

Related information

The Open Group: CIMIndicationProvider documentation

Troubleshooting the CIM server

Use this information if the CIM server does not start or if the CIM server starts, but does not run as expected.

The CIM server does not start

If the CIM server does not start, follow these steps:

1. Ensure that the correct options and product are installed on your system.
2. Ensure that the CIM server is configured correctly.

The CIM server does not run as expected

If you have trouble with the CIM server, follow these steps:

1. Check whether the certificate is expired.

   To check whether the CIM server is running, type WRKACTJOB JOB(QUMECIMOM) at a command line. If there is no active job, type the STRTCPSVR *CIMOM command to start a server.
2. Check whether the CIMOM repository is corrupted.

   Verify whether the repository directory and configuration files exist in the /QOpenSys/QIBM/UserData/UME/Pegasus/ directory of the integrated file system. If any of these files are missing, restore all the repository directories and files from your backup. If a backup does not exist, follow the instructions in “Restoring corrupted files” on page 24.
3. Verify whether you are attempting to process a request when the provider is not registered or enabled:
   a. Type cimprovider -l -s to list the name and status of the registered provider modules.
   b. Type cimprovider -l-m *module-name* to see the individual providers in that module.
4. Check the job log file.
   a. Type WRKACTJOB at a command line.
   b. Check the QSYSWRK subsystem to find the QUMECIMOM job.
   c. Select `5` (Work with), and then type `10` (Display job log, if active, on job queue, or pending).
   d. If the QUMECIMOM job is not running, type WRKJOB QUMECIMOM.
   e. Select the most recent job by typing `1` (Select) next to it.
   f. If the status is OUTQ, type `4` (Work with spooled files), and then type `5` (Display) next to the QPJOBLOG file.

Related information for Common Information Model

Web sites and other information center topic collections contain information that relates to the Common Information Model topic collection. You can view or print any of the PDF files.

Web sites

- Web-Based Enterprise Management (WBEM) (http://www.dmtf.org/standards/wbem)

  The site is the official home of the Web-Based Enterprise Management (WBEM) initiative.
- Common Information Model: Introduction to CIM (http://www.wbemsolutions.com/tutorials/CIM/cim.html)

  This site provides a tutorial of CIM.
- The Open Group: OpenPegasus (http://www.openpegasus.org)

  This is the OpenPegasus home page.

Other information

- Network authentication service
- Host name resolution considerations
- Managing keytab files
- Backing up your system
- Digital Certificate Manager (DCM)

Related reference

“PDF file for Common Information Model” on page 3
You can view and print a PDF file of this information.

Appendix. 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 grant 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.

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

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.

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.

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

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

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
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, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Some parts of this document were included from the web site openpegasus.org.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This Common Information Model (CIM) publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of the IBM i5/OS.

Trademarks

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

i5/OS
IBM
IBM (logo)
PowerPC
System i

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Printed in USA