IBM i
7.2

System management
Common Information Model

Note

Before using this information and the product it supports, read the information in "Notices" on page 321.

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 IBM i 7.2 and later, CIM is a feature of the IBM Universal Manageability Enablement for i licensed program (5770-UME V1R4M0). 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, V2.2 and CIM operations over HTTP, V1.1. For more information about these standards, see the WBEM Web site.

CIM on IBM i 7.2 and later environment and dependence considerations

- CIM Server runs as a single TCP/IP server job in the QSYSWRK subsystem. There is only one instance of CIM Server. CIM Server itself is multi-threaded (using pthread library). The providers run in separate jobs (Out of process support required). 5770-UME V1R4M0 CIM server is configured as an auto-start service when doing scratch install of IBM i 7.2 and later system. IBM Navigator for i can be used to enable/disable the auto-start with TCP/IP feature.
- CIM Server enables the SSL connections by default. The CIM server checks for its private key and certificate during startup, and if either of those files does not exist, will create its private key and a self-signed 365 day certificate.
- CIM server depends on OpenSSL command to generate the certificates. 5733SC1 (OpenSSL) and 5770SS1 Option 33 (PASE) are the pre-request licensed programs of 5770-UME V1R4M0. The start-up code of CIM server checks the existence of these dependency licensed programs and give an error message if they are not there.

The IBM i 7.2 and later operating systems support only the IBM Universal Manageability Enablement for i licensed program (5770-UME V1R4M0). The 5770–UME V1R4M0 licensed program is also supported on IBM i 6.1 and IBM i 7.1

Related information

Web-Based Enterprise Management (WBEM)

Common Information Model: Introduction to CIM

Common Information Model (CIM) Standards

The Open Group: OpenPegasus

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

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
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

The IBM licensed program (5770–UME, V1R4M0) is supported on IBM i 6.1 and later operating system. After you upgrade the licensed program to 5770–UME V1R4M0, or after you upgrade from the V5R4 system-based CIM function to the 5770–UME V1R4M0 licensed program, you must start the CIM server before you configure the Common Information Model Object Manager (CIMOM). 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.

Repository migration

The IBM Universal Manageability Enablement for i 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 to Distributed Management Task Force (DMTF) CIM schema V2.29 in 5770-UME V1R4M0. 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 PGS10080
This message is written into the CIM server log when the migration begins.

Message PGS10081
This message is written into the CIM server log when the migration is completed without errors.

Configuration migration

The CIM server migrates the configuration file. The configuration properties are migrated and their values are not changed. If it is migrated to 5770-UME V1R4M0, following configuration properties are changed or not migrated:

- logdir, home, daemon, slp, repositoryDir, tempLocalAuthDir, and kerberosServiceName and enableHttpLocalConnection are not migrated from operating system CIM server in IBM i V5R4.
- If the value of property httpAuthType in the operating system CIM server is Kerberos, then enableHttpsConnection, sslClientVerificationMode and httpAuthType are not migrated. Property enableHttpConnection is set to false after the migration.
- exportSSLTrustStore, enableSSLExportClientVerification, httpBindAddress, httpsBindAddress, tempLocalAuthDir, enableClientCertification, httpAuthType, passwordFilePath, enableHttpExportConnection, httpExportPort, messageDir and providerDir are not migrated from 5722-UME V1R2M0.
- If the value of idleSessionTimeout is valid, property idleConnectionTimeout is set as the same value of idleSessionTimeout, and idelSessionTimeout is not migrated.
- If previous 5770-UME version is lower than V1R3M0 and traceLevel=4, then modify it to traceLevel=5.

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.

Preparing to use CIMOM

When you install the IBM i operating system, the IBM Universal Manageability Enablement for i is installed on the system by default. If an earlier version exists, the CIM Server will migrate the repository and configuration file from an early version. 5770-UME V1R4M0 CIM server is configured as an auto-start service when doing a scratch installation of IBM i 7.2 and later system. When the CIM server starts up, the configuration in the migrated configuration file will be loaded.

Note: If you upgrade the IBM i operating system from V5R4, the CIM function that is supplied with IBM i V5R4 is disabled. If you use the system-based CIM function in V5R4, when the upgrade finished, you must start the CIM server before you configure CIMOM. After upgrade from lower version to 5770UME V1R4M0, 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 2 for details.

Ensuring that the operating system has the required installation requirements

CIM requires specific installation options and product on the IBM i operating system.

You must have the following programs installed on the system:

- IBM i Portable Application Solutions Environment (IBM i PASE) (5770-SS1, option 33)
- OpenSSH, OpenSSL, zlib functions, IBM Portable Utilities for i (5733-SC1, option 1)
- IBM Universal Manageability Enablement for i (5770-UME V1R4M0)

Note: In IBM i 6.1 or 7.1, you need to install all the required fixes. Refer to Info APAR (5770-UME V1R4M0 CONSIDERATIONS).

Setting the configuration properties

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

You can use the **`cimconfig`** command with the `-s` option to set the current or planned configuration properties. To change the planned value of the CIM server, you use the `-p -s` options. To change the current value of the CIM server, you use the `-c -s` options.

To configure CIMOM, follow these steps:

1. "Ensuring that the operating system has the required installation requirements" on page 3
2. Set the configuration properties by using the **`cimconfig -p -s`** or **`cimconfig -c -s`** 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. Restart **CIMOM**

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: IBM 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 IBM i Portable Application Solutions Environment (IBM i 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. With this property, you can be authenticated through certificate verification or basic authentication. The sslTrustStore property gives the location of the truststore.

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	false	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.
enableHttpsConnection	true	This property enables the HTTPS port to listen for HTTPS requests.
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.
httpsPort	5989	This property specifies the port number of HTTPS requests that server listens on. You should set the value to a valid port number. This property only takes effect when the enableHttpsConnection property is set to true.

Related concepts

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.

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*, *traceFilePath*, *enableAuditLog*, *enableIndicationService*, *enableNormalization*, *socketWriteTimeout*, *traceFacility*, *maxLogFileSizeKBytes*, and *maxFailedProviderModuleRestarts* 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 for these changes to 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). This property applies to requests on the main SSL port.
enableAssociationTraversal	No	true	This property enables the association traversal. You can set the value to true or false.
enableAuditLog	Yes	false	This property enables or disables audit logging at run time.
enableIndicationService	Yes	true	This property enables the indication service. You can set the value to true or false.
enableNormalization	Yes	false	This property controls whether to normalize objects from trusted entities. This property works only 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.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
enableSubscriptionsForNonprivilegedUsers	No	true	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.
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.
idleConnectionTimeout	No	0	If set to a positive integer, this value specifies a minimum timeout value for idle client connections. If set to zero, idle client connections do not time out.
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.
repositoryIsDefaultInstanceProvider	No	true	This 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 IBM i providers that implement CIM metric classes no longer function properly.
socketWriteTimeout	Yes	20	This property defines the timeout (in seconds) for the socket on the server. If the CIM Server receives an EWOULDBLOCK/EAGAIN error on a non-blocking write, *socketWriteTimeout* defines the number of seconds the CIM Server will wait for the socket to get ready and resume writing data.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
sslCertificateFilePath	No	ssl/keystore/servercert.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* 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	ssl/keystore/serverkey.pem	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, and if the *sslTrustStore* property is set to empty, no certificates are trusted. If the *sslClientVerificationMode* property is set to disabled, this property is not used.
sslClientVerificationMode	No	optional	The property sets the mode of SSL client certificate verification. This property is effective only 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. **optional** The CIM server verifies a client certificate, if one is available. **disabled** The CIM server doesn't verify a client certificate.

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
sslTrustStoreUserName	No	QYCMCIMOM	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 IBM i 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	60	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.
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" on page 12.
traceFacility	Yes	File	This property defines the facility for trace messages. Valid values include: File, Log .

Table 2. Advanced startup properties (continued)

Property	Dynamic	Default value	Description
traceLevel	Yes	0	This property indicates the level of debug trace. If the value is 1, the program traces only function exits (the minimum trace). A trace level of 5 is the maximum trace. **Note:** If the *traceLevel* property is set to 5 and the *traceComponents* property is set to ALL, the size of the trace file grows quickly and uses a lot of disk space. The range of values is 1 to 5. If *traceLevel* is set to 4 in 5722-UME, then it will be modified to be set to 5 when updated to 5770-UME.
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.
traceMemoryBufferKbytes	No	10240	This property indicates the size of the of the memory trace buffer. The value is the allocated memory size in kilobytes (1024 bytes).
maxLogFileSizeKBytes	Yes	32768	This property specifies the maximum size of the log file in kilobytes. If the log file size exceeds the value of *maxLogFileSizeKBytes*, the log file will be pruned.
maxFailedProviderModuleRestarts	Yes	3	If set to a positive integer, this property specifies the number of times the failed provider module with indications enabled are restarted automatically before being moved to Degraded state. If this property is set to zero, the failed provider module is not restarted with indications enabled automatically and will be moved to Degraded state immediately.
maxIndicationDeliveryRetryAttempts	No	5	If set to a positive integer, this value defines the number of times that the indication service will try to deliver an indication to a particular listener destination. This does not effect the original delivery attempt, thus if set to 0, cimserver will only try to deliver the indication once.
minIndicationDeliveryRetryInterval	No	480	If set to a positive integer, this value defines the minimal time interval in seconds for the indication service to wait before retrying to deliver an indication to a listener destination that previously failed. Cimserver may take longer due to QoS or other processing.
sslCipherSuite	No	DEFAULT	String containing OpenSSL cipher specifications to configure the cipher suite the client is permitted to negotiate with the server during the SSL handshake phase.

Related concepts

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.

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.

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
- Authentication
- Authorization
- BinaryMessageHandler
- CimData
- CIMExportRequestDispatcher
- CIMOMHandle
- CMPIProvider
- CMPIProviderInterface
- Config
- ControlProvider
- CQL
- DiscardedData
- Dispatcher
- ExportClient
- Http
- IndHandler
- IndicationGeneration
- IndicationDeliveryRetry
- IndicationReceipt
- IndicationHandlerService
- IndicationService
- IndicationServiceInternal
- IPC
- L10N
- Listener
- LogMessages
- MessageQueueService
- ObjectResolution
- OsAbstraction
- ProviderAgent
- ProviderManager
- Registration
- Repository
- Server
- Shutdown
- Thread

- UserManager
- WQL
- WsmServer
- 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 0
Tracing is off

Level 1
Severe and log messages

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

Level 3
Interfunction logic flow, medium-data detail

Level 4
High-data detail

Level 5
High data detail, method entry and exit.

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 IBM i PASE shell and complete these steps:

1. Call qp2term.
2. Type the following commands:
   a. **cimconfig -s traceLevel=5**
   b. **cimconfig -s traceComponents=Thread, IndicationService**

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 IBM Navigator for i, or from a command-line interface; you can also configure the CIM server as an auto-start service.

Manual starting and stopping CIM Server

Note: During the start-up procedure, the CIM server starts a repository and configuration migration process and then creates the symbolic links. During this process, the CIM server is not available to process CIM requests. After that the CIM Server loads the repository and starts up with the configuration. If error found in migrated repository or configuration file, the CIM server will not start.

To start or stop a CIMOM job from IBM Navigator for i, follow these steps:

- Log on to *your system*.
- Click **IBM i Management** > **Network** > **Servers** > **User-Defined Servers**.
- On the User-Defined Servers 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.

Auto-start CIM Server

Following significant updates are introduced in 5770-UME V1R4M0.

5770-UME V1R4M0 CIM server is configured as an auto-start service when doing scratch install of IBM i 7.2, which means the CIM server can be started without any manual setup, as long as licensed program 5770-UME V1R4M0 is successfully installed. Successfully installation of licensed program 5770-UME V1R4M0 indicates all pre-requirements are met, so CIM server can run correctly. As a result of this, 5770-UME V1R4M0 CIM server will be automatically started along with every system start, such as IPL.

For systems which do not have licensed program 5770-UME V1R4M0 pre-installed, no matter whether it has been removed or not, CIM server does not start automatically after installation or re-installation of licensed program 5770-UME V1R4M0. It can only be started by running command **`STRTCPSVR *CIMOM`** manually, or starts along with restarting of the IBM i system.

Note: You can use IBM Navigator for i to configure the auto-start behavior of the CIM server by the following steps:

- From IBM Navigator for i, select **Network > Servers > User-Defined Servers**.
- Right click **CIMOM**, then select **Properties** (on the popup menu), enable/disable the auto-start feature by check/uncheck the item *Start when TCP/IP is started*.

CIMOM security

You have several options to ensure that the CIM server is secure on the IBM i 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
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
- 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
You can change basic startup properties for CIMOM with the **`cimconfig`** command.

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.

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.

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 IBM i system objects, but they change the CIM schema. For the IBM i operating system, Application Administration in IBM Navigator for i 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 IBM i Portable Application Solutions Environment (IBM i 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

IBM i Base Metrics Providers
DMTF management profile, DSP 1053 Base Metrics Profile, is implemented on IBM i. The Base Metrics Profile is a component profile that defines the minimum object model needed to provide dynamic metrics associated to existing managed elements and related associations.

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.

Working with authorization for CIM operations

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

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

1. From IBM Navigator for i, expand **Security** > **Application Administration**.
2. Select **Host Applications** > **CIMOM server**.
3. Add or remove a user or group's authorization to the following operations.
   - CreateClass
   - DeleteClass
   - DeleteQualifier
   - EnumerateClasses
   - EnumerateClassNames
   - EnumerateQualifiers
   - GetClass
   - GetQualifier
   - ModifyClass
   - SetQualifier

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 *** IOSYSCFG *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
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.

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

1. From IBM Navigator for i, expand **Security** > **Application Administration**.
2. Select **Host ApplicationsCIMOM server**
3. Select **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 *SECADM and *ALLOBJ). For the command-line interfaces that are not CIM clients, the special authority check is done in the command-line interface itself.

5770-UME V1R4M0 command-line commands and the authorization each command needs:

Commands	Authorization needed
cimconfig	*IOSYSCFG and *ALLOBJ
cimprovider	*IOSYSCFG and *ALLOBJ
cimtrust	*IOSYSCFG and *ALLOBJ and *SECADM
cimcrl	*IOSYSCFG and *ALLOBJ and *SECADM
cimsub	*IOSYSCFG and *ALLOBJ
cimmof	No special authorities are required to run this command
cimcli	No special authorities are required to run this command

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/repository.conf
  - /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/bin
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Events/data
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Events/lib
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mif
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mif/bin
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mif/data
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mib
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mib/bin
  - /QOpenSys/QIBM/UserData/UME/Pegasus/Mappings/Inventory/mib/data
- Restore Copyright Files:
  - /QOpenSys/QIBM/UserData/UME/Pegasus/aaa_README_COPYRIGHTS/Qume_Copyright_D
  - /QOpenSys/QIBM/UserData/UME/Pegasus/aaa_README_COPYRIGHTS/Qume_Copyright_D_I
  - /QOpenSys/QIBM/UserData/UME/Pegasus/aaa_README_COPYRIGHTS/Qume_Copyright_D_P
  - /QOpenSys/QIBM/UserData/UME/Pegasus/aaa_README_COPYRIGHTS/Qume_Copyright_P
- ProvData directory:
  - /QOpensys/QIBM/UserData/UME/Pegasus/ProvData
  - /QOpensys/QIBM/UserData/UME/Pegasus/ProvData/asset.dat
  - /QOpenSys/QIBM/UserData/UME/Pegasus/ProvData/rtMetricService.dat

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-3.6/lib/
- CIT spb file：/QOpenSys/QIBM/ProdData/UME/CIT/CIT_os400.spb
  - Licensed Program (LP) and Recovery used data info files:
  - /QOpenSys/QIBM/ProdData/UME/LPDEF
  - /QOpenSys/QIBM/ProdData/UME/LPDEF/LPDATAINFO.def
  - /QOpenSys/QIBM/ProdData/UME/LPDEF/LPRECOVER.def
  - /QOpenSys/QIBM/ProdData/UME/LPDEF/RepositoryData.def
- Licence file: /QOpenSys/QIBM/ProdData/UME/notices.txt

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

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/` and `bin/` directories is `PUBLIC *RX` or `PUBLIC *R`. 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` or `PUBLIC *R`. This directory only contains the IBM-included providers.

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

Table 3. Library objects (continued)

Library objects	Object type	Public authentication	Description
QUME/QCIMMSG	*MSGF	*USE	Message file
QUME/QUME0029	*PRDLOD	*USE	Machine readable information (MRI) product load
QUME/QUME0050	*PRDDFN	*USE	Product definition
QUME/QUME0050	*PRDLOD	*USE	Machine readable material (MRM) product load
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/QUMEUTIL	*SRVPGM	*USE	Audit log utility, *USE
QUME/QUMEUTILS	*SRVPGM	*EXCLUDE	Platform utilities

Adopted owner authority

The only object that adopt owner authority is *QUMECTLCIM *PGM*. The owner of this program is the *QSYS* object. The programs start and stop the CIM server.

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.*

Related concepts

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.

Related reference

Command authorization
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.

In 5770-UME V1R4M0, command line utility “cimreparchive” is shipped as an additional mechanism to backup repository. Please refer to “cimreparchive usage information ” on page 42 for details.

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 Systems Director.

Backup

It is recommended that the CIM server *UserData/* directory be backed up. Files in *UserData* are not expected to change very often in normal cases. They are mainly files that are created during set up, and are not changed. However, since applications or providers might change these files more often, a daily back up would be recommended. User may add content into repository or other files into *UserData*, the information will be mixed with IBM shipped information.

Note: The documentation requires distinguish between IBM files and User files in the *UserData/* directory, for it is not seem to be worth the same effort. The whole *UserData/* directory should be backed up. The customer could run the backup command to only save the changed files since the last backup.

Notes® about 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 SSL files in the UserData/ directory are not considered 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 considered 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, which would follow the recommended weekly backup. These are all “IBM” files that are installed with the UME licensed program, or PTF'd. All of these files can be restored from the backup media without special considerations.

Backup can be done by system command **SAV, SAVLIB**, some examples:

- *CRTSAVF FILE(TEST)* and *SAV DEV('/QSYS.lib/QGPL.LIB/TEST.FILE') OBJ(('/QOpenSys/QIBM/UserData/UME/Pegasus'))*
- *CRTSAVF FILE(TESTLIB)* and *SAVLIB LIB(QUME) DEV(*SAVF) SAVF(TESTLIB)*

Recovery

There are files in the CIM server's *UserData/* directory which are shipped as IBM data.

System management applications (Such as IBM Systems Director) may add User files to the CIM server's *UserData/* directory as part of their setup. The shipped IBM providers may add "User" files when they are called by the CIM server.

Finally, since the interface to change the CIM server's *UserData/* directory will be published, customers might want to add their own User data to this directory. Customers may configure the server, add SSL certificates, update the repository, etc.

Therefore, it is possible for the CIM server's *UserData/* directory to have a combination of IBM data and "User" data. In the CIM server design, the IBM data and "User" data are intermingled as files in the same directory, and sometimes as data in the same file.

The files that could contain both IBM and User data are the following:

1. All instance index files and instance data files. The most likely case of intermingled IBM and User data instances would be the provider registrations. Any providers developed by customers would be registered in the same files as the IBM shipped providers.
2. Instance association and class association files. These files keep track of associated classes or instances. Any associations created by customers would be stored in the same file as the IBM shipped associations.
3. Server configuration files.

The files which may contain both IBM and User data present a problem for recovery. A process that simply restores the IBM shipped file from the install media would recover the IBM data but would lose the User data. However, the goal is to recover the IBM data without destroying customer data. So to summarize: 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
You can use Application Administration to work with the authorization for CIM operations.

Related information

Backing up your system

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`**, **`cimtrust`**, **`cimcrl`**, **`cimsub`**, **`cimreparchive`** and **`cimcli`**.

For the IBM i implementation, the **`cimconfig`**, **`cimsub`**, and **`cimprovider`** commands require *IOSYSCFG and *ALLOBJ special authorities. The **`cimtrust`** and **`cimcrl`** command requires *ALLOBJ, *IOSYSCFG, and *SECADM special authorities. You do not need special authorities to run the **`cimmof`** and **`cimcli`** commands.

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 IBM i Portable Application Solutions Environment (IBM i PASE).

Related tasks

Configuring 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. This command requires *IOSYSCFG and *ALLOBJ authority to change config.

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

Related tasks

Configuring CIMOM

cimcrl usage information

You can use this command to manage X.509 certificates in a PEM CRL store. 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. This command requires **ALLOBJ, *IOSYSCFG* and **SECADM* authorities to manage certificates.

Name

`cimcrl`

Remove or list X509 certificates in a PEM CRL store.

Synopsis

Usage:

- `cimcrl -a –f file`
- `cimcrl -r -i issuer`
- `cimcrl -l [ -i issuer ]`
- `cimcrl --help`
- `cimcrl --version`

Description

The **cimcrl** command provides a command line interface to manage X509 CRLs in the *CRL* store. The *CRL* store contains the CRLs used to revoke certificates in the CIM Server truststore. For more information on *truststore* operations see the **cimtrust** command.

The *add* option of the **cimcrl** command adds the CRL from file to the *CRL* store. If a CRL for the specified issuer already exists in the *CRL* store, the CRL is replaced with the specified CRL.

The *remove* option of the **cimcrl** command removes the CRL for a specified issuer from the *CRL* store.

The *list* option of the **cimcrl** command lists the CRLs in the *CRL* store. The list can be filtered by specifying an issuer.

Options

-a

Add the specified CRL to the *CRL* store. If the specified file contains an invalid CRL, an error message is returned and no action is taken. If the CRL for the specified issuer already exists in the *CRL* store, the CRL is replaced with the CRL specified in file.

-r

The remove option removes the CRL from the *CRL* store for a specified issuer.

-l

Display the CRLs in the CRL store

-f file

Specify a PEM format file containing a CRL.

-i issuer

Specify the issuer name for a CRL.

--help

Display the command help message.

--version

Display the CIM Server version number.

Remarks

The **cimcrl** command requires that the CIM Server is running. This command operates on a CRL store on the local system only.

Exit Status

When an error occurs, an error message is written to stderr and an error value is returned. The following values are returned:

0	Success
1	General error
2	Connection failed
3	Connection timeout
4	CRL does not exist

Usage Notes

In 5722-UME **ssltrustmgr** command is used to provide an interface to manage X509 certificates in a truststore or X509 Certificate Revocation Lists in a *CRL* store. In 5770-UME V1R4M0, it will be split into *truststore* management and CRL management functionality with two separate commands **cimtrust** and **cimcrl.**

The command must be run from a PASE command line, which requires that the PASE product is installed on the system. It also requires that the CIM Server is running. You can run this command from */QOpenSys/usr/bin* directory.

Examples

`cimcrl -a -f class1crl.pem`

Add the CRL in *class1crl.pem* to the *CRL* store, .

`cimcrl –l -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com"`

List the CRL for a specified issuer.

`cimcrl –r -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com"`

Remove the CRL for a specified issuer.

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.

Remarks

Only a user with authorizations to perform the operations done by **cimmof** can run this command. This authorization is enforced by the CIM server, and will be different depending on the objects being compiled (classes, qualifiers, instances) and the namespaces in which they are being compiled. An appropriate message is returned by the server if specific authorities are needed.

Super classes must be compiled before subclasses, else the compile will fail.

It is strongly recommended that MOF files include all necessary sub-classes, so they can compile properly even if certain classes are not in the CIM Repository.

Diagnostics

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

——The CIM Server is not running. Start the CIM Server with the command and re-run **cimmof.**

If the MOF Compiler detects an error in the MOF file while parsing the file, a parsing error is returned with the line number of the MOF file containing the error. Operation cannot be carried out since the specified super class does not exist.

——The MOF Compiler compiled a MOF file with super classes that were not in the CIM Repository.

For a list of possible error messages that may be returned, refer to the Chapter on WBEM messages in the Administrator's Guide.

EXIT STATUS

The **`cimmof`** command returns one of the following values:

`0`

Success

`1`

Error

Usage Notes

When execute **`cimmof`** command without an input MOF file name, it will try to read *STDIN. In QP2TERM* there is no way to input *EOF* in *STDIN*, so the program will never return. In that case, it's needed to exit *QP2TERM* or kill the process in another session. So it's strongly recommended that using the command with an input MOF file or using it in a term that can get *EOF* input

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.

cimsub usage information

You can use this command to manage CIM indication subscriptions on the local CIM Server. 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. This command requires *ALLOBJ and *IOSYSCFG authorities to manage certificates.

Name
`cimsub`

Create, enable, disable, or remove a subscription, display selected subscription information, as well as create, remove, and display filters and handlers.

Synopsis
Usage:

- `cimsub -c f [fnamespace:]filtername -Q query [-L querylanguage] [-N sourcenamespace]`
- `cimsub -c h [hnamespace:][hclassname.]handlername [-D destination]| [-T snmptargethost [-P snmpportnumber] -V snmpversion [-S snmpsecurityname] [-E snmpengineid]]`
- `cimsub -c s [-n namespace] -F [fnamespace:]filtername -H [hnamespace:][hclassname.]handlername`
- `cimsub -l s|f|h [-v] [-n namespace] [-F [fnamespace:]filtername] [-H [hnamespace:][hclassname.]handlername]`
- `cimsub -e [-n namespace] -F [fnamespace:]filtername -H [hnamespace:][hclassname.]handlername`
- `cimsub -d [-n namespace] -F [fnamespace:]filtername -H [hnamespace:][hclassname.]handlername`
- `cimsub -r s|f|h|a [-n namespace] [-F [fnamespace:]filtername] [-H [hnamespace:][hclassname.]handlername]`
- `cimsub --help`
- `cimsub --version`

Description
The **`cimsub`** command provides a command line interface to manage CIM indication subscriptions on the local CIM Server.

- The first three forms of the **`cimsub`** command create the specified subscription, filter, handler, or subscription and its referenced filter and handler.
- The fourth form of the **`cimsub`** command lists all or selected indication subscriptions, filters, or handlers, and displays the requested information about the instance(s).
- The fifth form of the **`cimsub`** command enables the specified subscription, i.e. the subscription instance is modified to set the value of the *SubscriptionState* property to Enabled. When a subscription is enabled, the CIM Server attempts to process the subscription if at least one provider is available to serve the subscription.
- The sixth form of the **`cimsub`** command disables the specified subscription, i.e. the subscription instance is modified to set the value of the *SubscriptionState* property to Disabled. When a subscription is disabled, the CIM Server does not attempt to process the subscription, regardless of whether any providers are available to serve the subscription.
- The seventh form of the **`cimsub`** command removes the specified subscription, filter, handler, or subscription and its referenced filter and handler, i.e. each instance is deleted from the repository, and the CIM Server will no longer have any information about the instance(s). A filter or handler may not be removed if it is referenced by any other subscription.

Options

-c
Create specified subscription, filter, and handler (CIM_ListenerDestinationCIMXML, if not specified hclassname).

-l
List and display information about all or selected indication subscriptions ("s"), filters ("f"), or handlers ("h"). It is an error to specify the `-F` option with `-lh`. It is an error to specify the `-H` option with `-lf`. With the `-lf` or `-lh` option, *fnamespace* or *hnamespace* takes precedence over the -n namespace option if both are specified.

-e
Enable the specified subscription (set the *SubscriptionState* to Enabled).

-d
Disable the specified subscription (set the *SubscriptionState* to Disabled).

-r
Remove the specified subscription ("s"), filter ("f"), handler ("h"), or subscription and its referenced filter and handler ("a") (delete the instance(s) from the repository). The `-F` and `-H` options are required with -rs or -ra. It is an error to specify the `-F` option with `-rh`. It is an error to specify the `-H` option with `-rf`. With the `-rf` or `-rh` option, *fnamespace* or *hnamespace* takes precedence over the `-n` namespace option if both are specified.

-v
Include verbose information (e.g. SubscriptionState value, filter query, handler destination) in the information displayed for each listed instance.

-n namespace
Specify the namespace. For the `-l` forms of the command, if no namespace is specified, instances in all namespaces are listed. For all other forms of the command, if no namespace is specified, the command operates on an instance in the namespace *root/PG*_InterOp.

-F [fnamespace:]filtername
Specify the Filter Name. Omission of the filter namespace specifies that it is the same as that of the subscription.

-H [hnamespace:][hclassname.]handlername
Specify the Handler Name. Omission of the handler namespace specifies that it is the same as that of the subscription. Omission of the handler classname specifies that it is *CIM_ListenerDestinationCIMXML.*

--help
Display the help message.

--version
Display the CIM Server version.

Remarks
The **cimsub** command requires that the CIM Server is running

Exit Status
When an error occurs, an error message is written to stderr, and an error value is returned. The following values are returned:

0	Success
1	General error
2	Connection failed
3	Connection timeout
4	Access was denied
5	Namespace does not exist
6	Object could not be found
7	Operation is not supported

Usage Notes
The **cimsub** command requires the CIM server to be running. If the configuration property *enableSubscriptionsForNonprivilegedUsers* is set to false, only a privileged user may list, enable, disable, or remove a subscription, filter, or handler. If the *nableSubscriptionsForNonprivilegedUsers* configuration property is set to true, only a privileged user or the instance creator may enable, disable, or remove a subscription, filter, or handler. Caution should be used when specifying the -r option, since subscriptions, filters, or handlers removed may not be easily re-created, and may not be re-created using this command.

Examples

cimsub -cf test/TestProvider:filter1 -Q 'select * from IBMi_CacheBatteryEvent where DaysToError < 55'
Create a filter. The `-Q` option specifies the select condition for the event.

cimsub -ch root/ibmsd:handler1 -D https://ServerName:Port/CIMListener/DirectorConsumer
Create a handler. The `-D` option indicates where the event should be sent.

cimsub -cs -n root/ibmsd -F test/TestProvider:filter1 -H test/TestProvider:handler1
Create a subscription. The user should have the listener available to wait for the event. The event is organized as a CIM instance and the user can pick up properties from the instance.

cimsub –ls
List all indication subscriptions in all namespaces on the local CIM Server.

cimsub -ls -F TestFilter
List all indication subscriptions referencing the Filter with Name *TestFilter* in all namespaces on the local CIM Server.

cimsub -ls -H TestHandler
List all indication subscriptions referencing the *CIM_ListenerDestinationCIMXML* Handler with Name *TestHandler* in all namespaces on the local CIM Server.

cimsub -ls -v -n root/cimv2
List all indication subscriptions in the `root/cimv2` namespace on the local CIM Server, including subscription state, filter query and handler destination (verbose listing output).

`cimsub -ls -v -n root/PG_InterOp -F TestFilter -H TestHandler`
Display information about the subscription referencing Filter with Name *TestFilter* and referencing *CIM_ListenerDestinationCIMXML* Handler with Name *TestHandler* (all in namespace `root/PG_InterOp`), including subscription state, filter query, and handler destination (verbose listing output).

`cimsub -lf -v -F root/cimv2:TestFilter`
Display information about the Filter with Name *TestFilter* in the `root/cimv2` namespace (verbose listing output).

`cimsub –lh`
IList all handlers in all namespaces on the local CIM Server.

`cimsub -e -F TestFilter -H TestHandler`
Enable the subscription referencing Filter with Name *TestFilter* and referencing CIM_ListenerDestinationCIMXML Handler with Name *TestHandler* (all in namespace `root/PG_InterOp`).

`cimsub -d -n root/cimv2 -F test1/cimv2:TestFilter -H test2/cimv2:PG_ListenerDestinationSystemLog.TestHandler`
Disable the subscription in namespace `root/cimv2,` referencing Filter with Name *TestFilter* in namespace `test1/cimv2`, and referencing *PG_ListenerDestinationSystemLog* Handler with Name *TestHandler* in namespace `test2/cimv2`.

`cimsub -ra -n root/cimv2 -F TestFilter -H TestHandler`
Remove the subscription referencing Filter with Name *TestFilter*, and referencing *CIM_ListenerDestinationCIMXML* Handler with Name *TestHandler* (all in namespace `root/cimv2`), also removing the filter and handler instances (if not referenced by any other subscription).

cimtrust usage information

You can use this command to manage X.509 certificates in a PEM trust store. 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. This command requires **ALLOBJ, *IOSYSCFG* and **SECADM* authorities to manage certificates.

Name
`cimtrust`

Remove or list X509 certificates in a PEM format trust store.

Synopsis
Usage:

- `cimtrust -a [ -U certuser ] -f file –T ( a | e | s )`
- `cimtrust -r -i issuer ( -n serialnumber | -S subject )`
- `cimtrust -l [ -i issuer [ -n serialnumber | -S subject ] ]`
- `cimtrust -help`
- `cimtrust -version`

Description
The add option of the **cimtrust** command adds an X509 certificate file of a specified type to the truststore. The certuser specifies the username to be associated with the certificate in the file. If no certuser is specified, the certificate may not be used to authenticate a user.

The remove option of the **cimtrust** command removes the X509 certificate(s) matching the specified issuer and either serial number or subject from the *truststore*.

The list option of the **cimtrust** command lists the X509 certificates in the *truststore*. The list can be filtered by issuer and either serial number or subject.

Certificates in the trust store may be revoked by adding a Certificate Revocation List to the *CRL* store. For more information on *CRL* operations see the **cimcrl** command.

Options

-a
Add a certificate to the *truststore*. If the specified file does not contain a valid X509 certificate an error message is returned and no action is taken. If the X509 certificate already exists in the truststore, an error message is returned.

-r
Removes certificate(s) from the *truststore*. If the *truststore* contains multiple certificates matching the specified issuer and subject, all the matching certificates are removed. If there is no certificate for the specified issuer and either serial number or subject, an error message is returned and no action is taken.

-l
Display the X509 certificates in the *truststore*.

-f file
Specify a PEM format file containing an X509 certificate.

-U certuser
Specify a username to be associated with the specified certificate. The username specified should be a valid system user on the target system.

-i issuer
Specify the issuer name for a certificate.

-n serialnumber
Specify a certificate serial number.

-S subject
Specify the subject name for a certificate.

-T
Specify the type of a certificate. The type must be one of the following: authority (a): *root/intermediate* authority certificates. Certificates of this type are added to the trusted certificate store. The *certuser* is optional for authority certificates. If no *certuser* is specified, the certificate may not be used to authenticate a user. Authority issued end-entity (e): Certificates of this type are not added to the trusted *certificate* store. The *certuser* is required for authority issued end-entity certificates. Self-signed identity certificate (s): Certificates of this type are not added to the trusted certificate store. The *certuser* is required for self-signed identity certificates.

-help
Display the command help message.

-version
Display the CIM Server version number.

Remarks

The **cimtrust** command requires that the CIM Server is running. This command operates on a trust store on the local system only.

Exit Status

When an error occurs, an error message is written to stderr and an error value is returned. The following values are returned:

0	Success
1	General error
2	Connection failed
3	Connection timeout
4	Certificate already exists
5	Certificate does not exist

6	Invalid system yser

Usage Notes

In 5722-UME **ssltrustmgr** command is used to provide an interface to manage X509 certificates in a trust store or X509 Certificate Revocation Lists in a CRL store. In 5770-UME V1R4M0, it will be split into *truststore* management and CRL management functionality with two separate commands **cimtrust** and **cimcrl.**

The command must be run from a PASE command line, which requires that the PASE product be installed on the system. It also requires that the CIM Server is running. You can run this command from */QOpenSys/usr/bin* directory

Examples

`cimtrust -a -U guest -f cert.pem –T s`
Add the X509 self-signed identity certificate in the *cert.pem* file and associate it to *certuser* guest. This certificate will be added to the trusted certificate store.

`cimtrust -a -f ca.pem –T a`
Add the X509 authority root CA certificate in the *ca.pem* file with no *certuser* association. This certificate will be added to the trusted certificate store but may not be used to authenticate a user.

`cimtrust -a -f user.pem –U pegasus –T e`
Add the X509 authority issued end-entity certificate in the user.pem file and associate it to certuser pegasus. This certificate may be used to authenticate user pegasus but will not be added to the trusted certificate store.

`cimtrust -r -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com" -n 01`
Remove the certificate matching the specified issuer and serial number from the trust store.

`cimtrust -r -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com" –S "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com”`
Remove the certificate(s) matching the specified issuer and subject from the trust store.

`cimtrust -l`
List all the X509 certificates in the trust store.

cimcli usage information

You can use this command to test and query CIM Servers. This command includes operations to query for namespaces and to get all instances in a namespace.A symbolic link in the `/QopenSys/usr/bin` directory for this command is provided. No special authorities are required to run this command. Authority to perform the actions of this command is checked by the normal authorization checks in the CIM server.

Name
`cimcli`

Send CIM operations to a CIM Server.

Synopsis
Usage:

- `cimcli [`*`command`*`] [`*`CIMObject`*`] [`*`options`*`] *[`*`extra_parameters`*`]`
- `cimcli -h | --help | -hc | -ho | -h `*`command`*

Description
The **`cimcli`** command executes single CIM operations.

Operations
[*command*]

The operation (command) defines the operation to be performed. The valid options vary by which command is being run.

?

Shows list of commands.

```
cimcli ?
```

a

(associators) Enumerates the classes or instances linked (associated) to *classname* or *instancename*.

```
cimcli a classname | instancename [options]
```

an

(associatorNames) Enumerates the names of classes or instances linked (associated) to *classname* or *instancename*.

```
cimcli an classname | instancename [options]
```

ci

(createInstance) Creates instance of *classname* with the specified parameters using a **name=value* extra parameter.

```
cimcli ci classname *[name=value] [options]
```

dc

(deleteClass) Deletes class specified by *classname*.

```
cimcli dc classname [options]
```

di

(deleteInstance) Deletes instance of *objectname* or delete instance of *classname* interactively. Specify the `*keyPropertyName=`*value* extra parameter for equal key property values.

```
cimcli di objectname
cimcli di classname *[keyPropertyName=value]
```

dq

(deleteQualifier) Deletes qualifier specified by *qualifiername*.

```
cimcli dq qualifiername [options]
```

ec

(enumerateClasses) Enumerates classes. An optional *classname* can be specified.

```
cimcli ec [classname] [options]
```

ei

(enumerateInstances) Enumerates instances of *classname*, where *classname* must be the name of a CIM class.

```
cimcli ei classname [options]
```

eq

(enumerateQualifiers) Enumerates all qualifiers.

```
cimcli eq [options]
```

gc

(getClass) Gets class of *classname*.

```
cimcli gc classname [options]
```

gi

(getInstance) Gets instance of *objectname*, or get instance of *classname* interactively by specifying an option. Specify the *PropertyName=*value* extra parameter for equal property values.

```
cimcli gi objectname *[PropertyName=value]
cimcli gi classname [options] *[PropertyName=value]
```

gp

(getProperty) Gets single property of *objectname*.

```
cimcli gp objectname [options]
```

gq

(getQualifier) Gets qualifier specified by *qualifiername*.

```
cimcli gq qualifiername [options]
```

im

(invokeMethod) Executes an extrinsic method specified by *methodname* on a class or instance specified by *objectname*. The parameters are supplied as *name*=*value* pairs.

```
cimcli im objectname methodname {name=value[,name=value,...]}
```

nc

(enumerateClassNames) Enumerates class names. An optional *classname* can be specified.

```
cimcli nc [classname] [options]
```

ni

(enumerateInstanceNames) Enumerates instance names of *classname*, where *classname* must be the name of a CIM class.

```
cimcli ni classname [options]
```

niall

(enumallInstanceNames) Enumerates all instance names of a specified namespace.

```
cimcli niall [options]
```

ns

(enumerateNamespaces) Enumerates all namespaces on the server.

```
cimcli ns [options]
```

mi

(modifyInstance) Modifies instance of *objectname* or *classname* with the specified parameters using a **name*=*value* extra parameter.

```
cimcli mi objectname | classname *[name=value]
```

r

(references) Enumerates the association classes or association instances linked to *classname* or *instancename*.

```
cimcli r classname | instancename [options]
```

rn

(referenceNames) Enumerates the names of association classes or association instances linked to *classname* or *instancename*.

```
cimcli rn classname | instancename [options]
```

son

(Turn on statistics) Turns on CIM Server statistics gathering.

```
cimcli son [options]
```

soff

(Turn off statistics) Turns off CIM Server statistics gathering.

```
cimcli soff [options]
```

sp

(setProperty) Sets single property of *objectname*.

```
cimcli sp objectname [options]
```

ti

(testInstance) Tests instance of *objectname* with the specified parameters using a **name=value* extra parameter, or test instance of *classname* while specifying the `*PropertyName=`*value* extra parameter for equal property values.

```
cimcli ti objectname *[name=value]
cimcli ti classname *[PropertyName=value]
```

xq

(execQuery) Performs the ExecQuery CIM operation with the specified *query-expression* and optional *query-language*.

```
cimcli xq query-expression [query-language]
```

Object

[*CIMObject*]

The name of the object on which the command should operate.

Options

[*options*]

Options are identified with the - or -- notation.

-ac

(assocClass) Defines an **assocClass** string for Associator calls. The default value is empty.

-ar

(assocRole) Defines a role string for an Associator's **assocRole** parameter. The default value is empty.

–cert

(clientCert) Specifies a client certificate file path to present to the server. This option is optional and has an effect only on connections made over HTTPS using `-s` with the default value of `false`.

-count

(count) Displays the expected count of objects returned if the `-sum` is specified. The `-count` option tests the count and displays the difference. A nonzero status code is returned if the test fails. The default value is 29346.

-d

(debug) Displays more detailed debug messages. The default value is `false`.

-delay

(delay) Specifies the delay seconds between connection and request. The default value is `0`.

-di

(deepInheritance) Sets the **deepInheritance** parameter to `true` on requests that honor this parameter. namespace for the operation. The default value is `false`.

-f

(filter) Defines a filter to use for a query. Specify input in a single string. The default value is empty.

–h
(help) Displays help usage message. The default value is `false`.

-hc
(help commands) Displays CIM Operation command list. The default value is `false`.

-–help
(full help) Displays full help message with commands, options, and examples. The default value is `false`.

-ho
(help options) Displays list of options. The default value is `false`.

-i
(interactive) Interactively prompts the user to select instances. Used with associator and reference operations. The default value is `false`.

-ic
(includeClassOrigin) Sets the **`includeClassOrigin`** parameter to `true` on requests that honor this parameter. namespace for the operation. The default value is `false`.

-iq
(includeQualifiers) Sets the **`includeQualifiers`** parameter to `true` on operations. The default value is `false`.

Note: `includeQualifiers` has been deprecated by the DMTF for some operations and will completely be removed with the next major version of CIM.

–key
(clientKey) Specifies a client private key file path. This option is optional and has an effect only on connections made over HTTPS using `-s` with the default value of `false`.

-l
(location) Specifies the host name for the CIM server and optionally the port (*HostName:port*). . The default value is (empty).

-lo
(localOnly) Sets the **`localOnly`** parameter to `true` on operations. The default value is `true`.

Note: `localOnly` has been deprecated by the DMTF for some operations and will completely be removed with the next major version of CIM.

-n
(namespace) Specifies the namespace for the operation. The default value is `root/cimv2`.

-niq
(notIncludeQualifiers) Sets the **`notIncludeQualifiers`** parameter to `false` on operations. The default value is `false`.

Note: `includeQualifiers` has been deprecated by the DMTF for some operations and will completely be removed with the next major version of CIM.

-nlo
(notLocalOnly) Sets the **`notLocalOnly`** parameter to `false` on operations. The default value is `false`.

Note: `localOnly` has been deprecated by the DMTF for some operations and will completely be removed with the next major version of CIM.

-o
(outputformats) Specifies an output format. Valid values are: `xml`, `mof`, `txt`, and `table`. The default value is `mof`.

-p
(password) Specifies the password for authentication. The default value is (empty).

-pl
(propertyList) Defines a **`propertyNameList`**. Format is *`p1,p2,p3`* (without spaces). Use blank (empty) for an empty list. The default value is empty.

-ql
(queryLanguage) Defines a query language to be used with a query filter. The default value is SQL.

-r
(role) Defines a role string for a reference **role** parameter. The default value is empty.

--r
(repeat) Specifies the number of times to repeat the function. 0 means one time. The default value is 0.

-rc
(resultClass) Defines a **resultClass** string for References and Associators. The default value is empty.

-rr
(resultRole) Defines a role string for an Associator's operation **resultRole** parameter. The default value is empty.

-s
(SSL) Specifies to connect over HTTPS. The default value is false.

--setRtnHostNames
(setRtnHostNames) Sets the namespace component of the reference and path outputs parameter. This option is used to allow comparison of paths and instances without involving the variable of host namespaces. The default value is false.

–sort
(sort) Specifies to sort the returned entities for multi-entity responses. The default value is false.

–sum
(summary) Specifies to display summary output. Displays counts of objects returned instead of the names or objects themselves. The default value is false.

-t
(time) Specifies to measure the time for the operation and to display the results upon command completion. The default value is false.

-trace
(traceLevel) Sets the common components trace level. 0 is "off". Valid values are 0 to 4. The default value is 0.

-u
(user) Specifies a user name for authentication. The default value is empty.

-v
(verbose) Specifies to display verbose output. Displays detailed parameter input and other request processing information. The default value is false.

–version
(version) Displays the software version. The default value is false.

-x
(xmlOutput) Specifies to output objects in XML format. The default value is false.

Examples

cimcli ni -n test/TestProvider TST_Person
Enumerate instance names of class TST_Person.

cimcli niall -n root/cimv2
Enumerate instance names of all classes under the namespace root/cimv2.

cimcli ei PG_ComputerSystem
Enumerate instances of class.

cimcli nc
Enumerate class names fromroot/cimv2.

cimcli ec -n root/cimv2
Enumerate classes from namespace root/cimv2.

`cimcli gc CIM_door -u guest -p guest`
Get class for user `guest` and password `guest`.

`cimcli gi -n test/TestProvider TST_Person`
Get instance of class.

`cimcli ci -n test/TestProvider TST_Person name=Mike SSN=333`
Create instance of class TST_Person with properties of `name= Mike` and `SSN= 333`.

`cimcli ti -n test/TestProvider TST_Person name=Mike SSN=333`
Test instance of class for equality of Name and SSN properties. Returns error code if instance in server does not have the same properties and values as the instance built from input.

`cimcli di -n test/TestProvider TST_Person`
Delete instance of class TST_Person interactively.

`cimcli di -n test/TestProvider TST_Person name=Mike`
Delete instance of class TST_Person, with key property of `name= Mike`, using instance builder format to define object path.

`cimcli di -n test/TestProvider TST_Person.name=\"Mike\"`
Delete instance of class TST_Person, with object path `TST_Person.name="Mike"`, using object path input format.

`cimcli mi -n test/TestProvider TST_Person.Id=\"Mike\" SSN=444`
Modify the instance, if it exists, using rules of the DMTF modifyInstance operation.

`cimcli dc -n test/TestProvider TST_Person`
Deletes the class when there are no instances and sub-classes for this class.

`cimcli gp -n test/TestProvider TST_Person.Id=\"Mike\" SSN`
Gets the property named SSN from the instance defined by `TST_Person.Id="Mike"`.

`cimcli sp -n test/TestProvider TST_Person.Id=\"Mike\" SSN=333`
Sets the property named SSN from the instance defined by `TST_Person.Id="Mike"`.

`cimcli gq Association`
Gets the qualifier named `Association` in mof output in the default namespace (normally `root/cimv2`)

`cimcli eq -n test/TestProvider`
Enumerates qualifiers of namespace `test/TestProvider`.

`cimcli dq -n test/TestProvider ASSOCIATION`
Deletes qualifier named `ASSOCIATION` in namespace `test/TestProvider`.

`cimcli a TST_Person.name=\"Mike\" -n test/TestProvider -ac TST_Lineage -rc TST_Person`
Gets associations for instances of association class TST_Lineage and result class TST_Person associated with instance of TST_Person, with key `"Mike"` in namespace `test/TestProvider`.

`cimcli a TST_Person -n test/TestProvider -ac TST_Lineage`
Gets associated classes for all instances of classes associated with class TST_Person and with association class TST_Lineage.

`cimcli a TST_Person -n test/TestProvider -ac TST_Lineage -i`
Gets associated instances by prompting interactively for classes associated with class TST_Person and with association class TST_Lineage.

`cimcli an TST_Person -n test/TestProvider -ac TST_Lineage`
`cimcli r TST_Person.name=\"Mike\" -n test/TestProvider -rc TST_Lineage`
Gets references for instances of association class TST_Lineage associated with instance of TST_Person with key `"Mike"` in namespace `test/TestProvider`.

`cimcli r TST_Person.name=\"Mike\" -n test/TestProvider`
Gets references for all instances of classes associated with instance of TST_Person with key `"Mike"` in namespace `test/TestProvider`.

cimcli r TST_Person -n test/TestProvider -rc TST_Lineage

Gets reference classes for association class TST_Lineage and subclasses in namespace `test/TestProvider`.

cimcli rn TST_Person.name=\"Mike\" -n test/TestProvider -rc TST_Lineage

Gets referenceNames for instances of association class TST_Lineage associated with instance of TST_Person with key `"Mike"` in namespace `test/TestProvider`.

cimcli rn TST_Person.name=\"Mike\" -n test/TestProvider

Gets referenceNames for all instances of classes associated with instance of TST_Person with key `"Mike"` in namespace `test/TestProvider`.

cimcli rn TST_Person -n test/TestProvider -rc TST_Lineage

Gets referenceNames for class TST_Lineage in namespace `test/TestProvider`.

cimcli im Sample_MethodProviderClass.Name=\"mooo\" SayHello -n root/SampleProvider -ip p1=fred

Executes the method named SayHello on the instance Sample_MethodProviderClass.Name=\"mooo\" in namespace `root/SampleProvider`. The value of input parameter P1 of the method is `fred` .

cimcli ns

Enumerates all namespaces in repository.

cimcli son

Switches on the statistical information on CIM Server.

cimcli soff

Switches off the statistical information on CIM Server.

cimreparchive usage information

cimreparchive is a command generally used in OpenPegasus, well known in various platforms. It is supported as an additional mechanism to backup repository besides IBM i specific SAVE and RESTORE mechanism. A symbol link of it is added in directory */QOpenSys/usr/bin*. This command requires *NONE authority to run, but the user should have the accessibility authority of directory: "*/QOpenSys/QIBM/UserData/UME/Pegasus/repository*".

Name

`cimreparchive`

Backup repository.

Synopsis

Usage:

- `cimreparchive archive_file`
- `cimreparchive --version`
- `cimreparchive --help`

Description

The cimreparchive utility packages repository files that reside in directory "*/QOpenSys/QIBM/UserData/UME/Pegasus/repository*" into specific archive file.

Options

archive_file

The archive file path and name

--version

Display CIM Server version number

--help

Display this help message

Usage Notes

The cimreparchive command can be used whether the CIM Server is running or not. The user could use "pax -r –f archive_file" to restore the repository.

Dependency Considerations

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

SNMP Not Supported

The IBM Universal Manageability Enablement for i does not use SNMP (which is supported in 5722-UME). The following file is obsolete and will be deleted.

- **QUME/QUMESTRSA**: Starts and stops the SNMP sub-agent
- libCim2Snmp.so
- libCim2SnmpBridge.so
- libSnmpTrap.so

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.

The configuration property *slp* is used to enable or disable the registration with SLP service agent. The value of property *slp* will decide whether CIM server will register with SLP SA automatically, and the default value is set to *true*. In 5770-UME V1R4M0 the property *slp* is not exported to user, user can not configure this property, and the value of property *slp* is true.

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, IBM i Portable Application Solutions Environment (IBM i PASE) also supports SSL certificate-based client authentication on CIM requests from CIM clients and supports the CIM exports carrying indication data.

- **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 2.29**

  Schema 2.29 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 2.21

Compatibility to V5R3 and V5R4 operating system CIM providers

Classes that are supplied with the IBM Universal Manageability Enablement for i 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 i licensed program compatible to V5R3 and V5R4 operation system CIM providers. In IBM i 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

Table 4. Class map in root/ibmsd and root/cimv2 (continued)

Class names in root/ibmsd	Class names in root/cimv2
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 IBM i client partitions

You can use the IBM Universal Manageability Enablement for i licensed program (5770-UME) on IBM i client partitions, with limitations. Physical hardware information is not accessible to IBM i 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.
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 IBM i Portable Application Solutions Environment (IBM i 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

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

Provider name	Implements CIM class	Provider type	Category
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
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

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

Provider name	Implements CIM class	Provider type	Category
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.

Related reference

IBM_ComputerSystem
This provider is the same as IBMPSG_ComputerSystem.

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

Table 10. IBMPSG_BaseBoard (continued)

Property name	Property description	Value or value location
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*
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*

Table 11. IBMPSG_Chassis (continued)

Property name	Property description	Value or value location
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.	
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)

Related reference

IBM_Chassis
This provider is the same as IBMPSG_Chassis.

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*

Table 12. IBMPSG_FRU (continued)

Property name	Property description	Value or value location
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.	

Table 13. IBMPSG_PhysicalDisk (continued)

Property name	Property description	Value or value location
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 name of the scoping system.	
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)
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).	

Table 14. IBMPSG_PhysicalMemory (continued)

Property name	Property description	Value or value location
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.	
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.	

Table 14. IBMPSG_PhysicalMemory (continued)

Property name	Property description	Value or value location
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).	

Related reference

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.

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.	
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

Table 15. IBMPSG_PhysicalNetworkAdapter (continued)

Property name	Property description	Value or value location
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.	
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.	

Table 16. IBMPSG_Port (continued)

Property name	**Property description**	**Value or value location**
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
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.	

Table 16. IBMPSG_Port (continued)

Property name	Property description	Value or value location
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.
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

Table 17. IBMPSG_Processor (continued)

Property name	Property description	Value or value location
string SystemName (key) (256)	The name of the scoping system.	
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)

Related reference

IBMi_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.

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*

Table 18. IBMPSG_SerialNumberInformation (continued)

Property name	**Property description**	**Value or value location**
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*

Providers that are required by Director and eServer for Management Central Inventory equivalence

These providers make IBM i proprietary instrumentation available to IBM Director and eServer™ management applications in an industry standard format.

These providers instrument the following Inventories:

Software inventory providers

The Software Inventory Profile(DSP1023) describes the CIM schema elements required to provide an inventory of installed BIOS, firmware, drivers, and related software in a managed system. This profile also describes the CIM schema elements required to represent the software that can be installed on a managed system.

Supported providers

Software Inventory Profile will be referred to model the following IBM i Software Objects:

- Installed Products: a list of the software products that are currently installed on the selected system. You can send these products to one or more endpoint systems or system groups and install them on those systems.
- Supported Products: a list of the software products that the selected system currently supports for the other systems that it manages in the network. For example, this list can contain products that are not installed on this system. A system that provides support typically orders the fixes and sends them to systems where the product is installed.
- Software Fix: Program Temporary Fix(PTF)
- Software fix group： PTF group

CIM_SoftwareIdentity will be extended to represent IBM i Fix, Fix Group and Software Products. The relationship between Software Products, Fixes and Fix groups is illustrated by the three sub classes inherited from CIM_SoftwareIdentity.

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

Table 19. Software inventory Providers

Provider name	Implements CIM class	Provider type
QUME_SystemSpecificCollectionProvider	IBM_SystemSpecificCollection	Instance
QUME_HostedCollectionProvider	IBM_HostedCollection	Instance & Association
QUME_SupportedSoftwareIdentityProvider	IBMi_SupportedSoftwareIdentity	Instance & Association
QUME_SoftwareProductProvider	IBMi_SoftwareProduct	Instance
QUME_InstalledSoftwareIdentityProvider	IBM_InstalledSoftwareIdentity	Instance & Association
QUME_TemporaryFixProvider	IBMi_TemporaryFix	Instance
QUME_TempFixGroupProvider	IBMi_TempFixGroup	Instance
QUME_FixInSoftwareProductProvider	IBMi_FixesInSoftwareProduct	Instance & Association
QUME_FixesInFixGroupProvider	IBMi_FixesInFixGroup	Instance & Association
QUME_RequisiteFixesProvider	IBMi_RequisiteFixes	Instance & Association
QUME_DependentFixesProvider	IBMi_DependentFixes	Instance & Association
QUME_RelatedFixGroupProvider	IBMi_RelatedFixGroup	Instance & Association
QUME_SoftwareRegisteredProfileProvider	IBMi_SoftwareRegisteredProfile	Instance
QUME_SoftwareElementConformsToProfileProvider	IBMi_SoftwareElementConformsToProfile	Instance & Association

IBM_SystemSpecificCollection

This provider will return a collection of supported software on the system when an enumerated list of instances is asked for, or will look up the resource based on the Software Name provided as the key under the InstanceID property.

Table 20. IBM_SystemSpecificCollection

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	IBM i Supported Software Collection
string Description	The Description property provides a textual description of the object	IBM i Supported Software Collection
string ElementName	A user-friendly name for the object.	Supported Software

Table 20. IBM_SystemSpecificCollection (continued)

Property name	Property description	Value or value location
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Supported Software Collection

IBM_HostedCollection

This provider will return association between System and software collection.

Table 21. IBM_HostedCollection

Property name	Property value and data source	Instance mapping rule
CIM_System REF Antecedent	instance of System	This should be 1 to 1 association
CIM_SystemSpecificCollection REF Dependent	Returns instance of supported software collection	between IBMOS400_ComputerSystem and IBM_SystemSpecificCollection

IBMi_SupportedSoftwareIdentity

This provider is used to model supported Products.

Table 22. IBM_SupportedSoftwareIdentity

Property name	Property value and data source	Instance mapping rule
CIM_Collection REF Collection	Returns all instances of supported software collection	This should be 1 to N association between IBM_SystemSpecificCollection and IBMi_SoftwareIdentity.
CIM_ManagedElement REF Member	Returns instance of Software Identity	

IBMi_SoftwareProduct

This provider will return instances of all known or defined products available on the system when an enumerated list of instances is asked for, or will look up the resource based on the software Name provided as the key under the InstanceID property.

Table 23. IBMi_SoftwareProduct

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	software product *InstanceID*
string Description	The Description property provides a textual description of the object	
string ElementName	A user-friendly name for the object	*InstanceID*
uint16 OperationalStatus[]	Indicates the current statuses of the element	0 Unknown, 2 Installed 6 Error, 15 Supported

Table 23. IBMi_SoftwareProduct (continued)

Property name	Property description	Value or value location
string StatusDescriptions[]	Strings describing the various OperationalStatus array values	Unknown, Installed, Error, Supported
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	
string Manufacturer	Manufacturer of this software	
string VersionString	A string representing the complete software version information	
string Languages[]	The language editions supported by the software	
string LoadID	Load IDs are 4 characters in length; for example, 2924 is the load ID for an English national language version (NLV)	
string ComplianceType	The compliance type determines the action taken when the value of the usage limit field is exceeded	
uint32 GracePeriod	The number of days after a product first exceeds its usage limit that a user has to obtain a new license key	
datetime GracePeriodExpirationDate	Grace Period Expiration Date	
datetime KeyExpirationDate	The date the license will expire	
datetime LastPeakDate	The date and time when the peak usage of the product or feature last occurred since the peak usage was reset to zero	
datetime LastUpdatedDate	The date and time when the usage limit was last updated	
string Level	The level identifier of the product for which information was returned. The format is Lxx. The returned value is blank for all products other than the operating system and Licensed Internal Code	
string LicenseKey	The license key for the product, license term, feature, and system	
string LicenseTerm	The extent of time the authorized usage limit for a product lasts	

Table 23. IBMi_SoftwareProduct (continued)

Property name	Property description	Value or value location
string MessageFile	The name of the message file that contains the messages which describe the product and its options	
string MessageIdentifier	A seven character alphanumeric identifier assigned to the message that describes the product option selected	
uint32 PeakUsage	The maximum number of license users that have accessed the product or feature at one time	
boolean Packaged	Packaged	TRUE
string ProcessorGroup	A processor group is the grouping of system model numbers by relative processor size	
uint32 ProcessorPeakUsage	The maximum processor usage count in hundreths of processors	
uint32 ProcessorUsageCount	The processor usage count is the number of hundreths of processors in the logical partition configured at the time the product was used. This field is set to 0 for products that do not have a processor usage type	
string RegistrationType	The registration type associated with the product	
sint32 ThresholdValue	The threshold indicates you want a message sent to the system operator message queue stating that a product or feature is reaching the usage limit	
uint32 UsageCount	The usage count for the product or feature at the time of the retrieve operation	
sint32 UsageLimit	The usage limit for this license	
string UsageType	The usage type associated with this license	
string VendorData	Information the vendor defined at Generate License Key time	

IBM_InstalledSoftwareIdentity

This provider will return instances of installed Software Products available on the system.

Table 24. IBM_InstalledSoftwareIdentity

Property name	Property value and data source	Instance mapping rule
CIM_System REF System	Returns instance of computer system	This should be 1 to N association between IBM_ComputerSystem and IBMi_SoftwareIdentity.
CIM_SoftwareIdentity REF InstalledSoftware	Returns instance of Software Identity	

IBMi_TemporaryFix

This provider will return instances of all Software Product Fixes available on the system when an enumerated list of instances is asked for, or will look up the resource based on the product temporary fix name provided as the key under the InstanceID property.

Table 25. IBMi_TemporaryFix

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	Program temporary fix *ElementName*
string Description	The Description property provides a textual description of the object.	Program temporary fix *ElementName*
string ElementName	A user-friendly name for the object	
datetime InstallDate	A datetime value that indicates when the object was installed. (FLD Loaded Status Date)	
uint16 OperationalStatus[]	Indicates the current statuses of the element	
string StatusDescriptions[]	Strings describing the various OperationalStatus array values	
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	
string VersionString	A string representing the complete software version information	
uint32 ActionPending	Whether a required action has yet to be performed to make this PTF active	
uint32 ActionRequired	Whether an action is required to make this PTF active when it is applied	
boolean CoverLetterStatus	Whether a cover letter exists for the PTF	

Table 25. IBMi_TemporaryFix (continued)		
Property name	**Property description**	**Value or value location**
string FixMinimumLevel	The indicator of the lowest level of the product on which this PTF can be installed	
string FixMaximumLevel	The indicator of the highest level of the product on which this PTF can be installed	
boolean FixIsReleased	Whether the PTF save file is available for distribution to another system	
uint32 IPLAction	The action to be taken on this PTF during the next IPL	
uint32 IPLRequired	An IPL is required to apply this PTF	
boolean OnOrderStatus	Whether the PTF has been ordered	
boolean SaveFileStatus	Whether a save file exists for the PTF	
string SaveFileLocation	The name of the library where the save file for the PTF is located	
uint32 ServerIPLRequired	Indicates whether or not a server IPL must be performed in order to activate the changes for the PTF	
string SupersededBy	The identifier of the PTF that has replaced this PTF	
string SupersedingFix	The identifier of the most recent supersede of this PTF that exists on the system	
string LoadID	The load ID of the product load for the PTF	
string ProductID	The product ID for the PTF for which information is requested	
string Option	The PTF is for this option of the product	
string TargetOSRelease	The earliest release of the operating system on which you can load and apply the PTF	

IBMi_TempFixGroup

This provider will return instances of all Software Temporary Fix group available on the system when an enumerated list of instances is asked for, or will look up the resource based on the Fix group Name provided as the key under the InstanceID property.

Table 26. IBMi_TempFixGroup

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	PTF group *ElementName*
string Description	The Description property provides a textual description of the object.	
string ElementName	A user-friendly name for the object	
uint16 OperationalStatus[]	Indicates the current statuses of the element	
string StatusDescriptions[]	Strings describing the various OperationalStatus array val	0 Unknown. 1 Not applicable. 2 Supported only. 3 Not installed. 4 Installed. 5 Error.
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	
string VersionString	A string representing the complete software version information	
uint16 FixGroupLevel	The current level of the PTF group	
string SpecialValue	Special value	

IBMi_FixesInSoftwareProduct

This provider will return association between a fix and a product that the fix belongs to.

Table 27. IBMi_FixesInSoftwareProduct

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF PartComponent	instance of Temporary Fixes	This should be 1 to N association between IBMi_SoftwareIdentity and IBMi_TemporaryFix
CIM_ManagedElement REF GroupComponent	Returns instance of installed software product	
uint64 AssignedSequence	0	

IBMi_FixesInfixGroup

This provider will return association between a fix and a fix group that the fix belongs to.

Table 28. IBMi_FixesInFixGroup

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF PartComponent	instance of TemporaryFixes	This should be 1 to N association between IBMi_TempFixGroup and IBMi_TemporaryFix
CIM_ManagedElement REF GroupComponent	Returns instance of Temporary Fix Group	
uint64 AssignedSequence	0	

IBMi_RequisiteFixes

This provider will return associated Requisite PTFs of current PTF.

Table 29. IBMi_RequisiteFixes

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF Antecedent	Returns all instances of fixes that are requisite for this fix	This should be 1 to N association between fixes.
CIM_ManagedElement REF Dependent	This fix.	
uint64 AssignedSequence	0	
uint32 TypeOfRequisite		
string Product		
string Release		
string Option		
string Feature		
boolean Conditional		
boolean Applicable		

IBMi_DependentFixes

This provider will return associated dependent PTFs of current PTF.

Table 30. IBMi_DependentFixes

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF Antecedent	This fix	This should be 1 to N association between fixes
CIM_ManagedElement REF Dependent	Returns instance of fixes that are dependent on this fix.	
uint64 AssignedSequence	0	
uint32 TypeOfDependent		
string Product		
string Release		
string Option		
string Feature		

IBMi_RelatedFixGroup

This provider will return related fix groups of current fix group.

Table 31. IBMi_RelatedFixGroup

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF Antecedent	Returns all instances of fix groups	This should be 1 to N association between fix groups.
CIM_ManagedElement REF Dependent	Returns instance of fix groups that relate to this fix group.	
uint64 AssignedSequence	0	

IBMi_SoftwareRegisteredProfile

This provider will return instances of registered profile for Software Inventory.

Table 32. IBMi_SoftwareRegisteredProfile

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object.	IBM i Software Registered Profile
string Description	textual description of the object	IBM i Software Registered Profile
string ElementName	A user-friendly name for the object	Software Registered Profile
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	Software Registered Profile
string RegisteredName	The name of this registered profile.	Distributed Management Task Force (DMTF): Software Inventory Profile
uint16 RegisteredOrganization	The organization that defines this profile.	2
string RegisteredVersion	The version of this profile.	1.0.0
uint16 AdvertiseTypes[]	This property signifies the advertisement for the profile information.	3 (Service Location Protocol (SLP))

IBMi_SoftwareElementConformsToProfile

This provider will return association between Software Inventory Profile and software identity.

Table 33. IBMi_SoftwareElementConformsToProfile

Property name	Property value and data source	Instance mapping rule
CIM_RegisteredProfile REF ConformantStandard	instance of Software Inventory Profile	This should be 1 to N association between IBMi_SoftwareRegisteredProfile and IBMi_SoftwareProduct
CIM_ManagedElement REF ManagedElement	Returns instance of Software Identity	

User and group inventory providers

The implementation of User and Group on IBM i will refer to DSP1034 - simple identity management profile and DSP1039 - role based authorization Profile.

Supported providers

This inventory will support the following operations:

- Browsing IBM i User and group profile and user contact information.
- Modify user or group profile properties.
- Create or delete user or group profiles on system.
- Add or remove users from group profile.
- Assign, change or remove role or privileges to user.

CIM_SoftwareIdentity will be extended to represent IBM i Fix, Fix Group and Software Products. The relationship between Software Products, Fixes and Fix groups is illustrated by the three sub classes inherited from CIM_SoftwareIdentity.

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

Table 34. User and group inventory Providers

Provider name	Implements CIM class	Provider type
QUME_IdentityProvider	IBM_Identity	Instance
QUME_UserAccountProvider	IBMi_UserAccount	Instance & Association
QUME_GroupProvider	IBMi_Group	Instance & Association
QUME_UserIdentityProvider	IBMi_UserIdentity	Instance
QUME_GroupIdentityProvider	IBMi_GroupIdentity	Instance & Association
QUME_UserContactIdentityProvider	IBMi_UserContactIdentity	Instance
QUME_UserContactProvider	IBMi_UserContact	Instance
QUME_MemberOfGroupProvider	IBMi_MemberOfGroup	Instance & Association
QUME_AccountManagementCapabilitiesProvider	IBM_AccountManagementCapabilities	Instance & Association
QUME_AccountManagementServiceProvider	IBM_AccountManagementService	Instance & Method
QUME_AccountOnSystemProvider	IBM_AccountOnSystem	Instance & Association
QUME_AccountElementCapabilitiesProvider	IBMi_AccountElementCapabilities	Instance & Association
QUME_RelatedFixGroupProvider	IBMi_RelatedFixGroup	Instance & Association

Table 34. User and group inventory Providers (continued)

Provider name	Implements CIM class	Provider type
QUME_OwningAccountElementProvider	IBMi_OwningAccountElement	Instance
QUME_ServiceAffectsAccountProvider	IBMi_ServiceAffectsAccount	Instance & Association
QUME_AccountManagementRegisteredProfileProvider	IBMi_AccountManagementRegisteredProfile	Instance
QUME_AccountElementConformsToProfile	IBMi_AccountElementConformsToProfile	Instance
QUME_ServicePrivilegeProvider	IBMi_ServicePrivilege	Instance & Association
QUME_RoleElementCapabilitiesProvider	IBMi_RoleElementCapabilities	Instance & Association
QUME_HostedRoleServiceProvider	IBMi_HostedRoleService	Instance & Association
QUME_CommonRoleProvider	IBMi_CommonRole	Instance
QUME_DedicatedRoleProvider	IBMi_DedicatedRole	Instance
QUME_PrivilegeProvider	IBM_Privilege	Instance
QUME_ConcreteRoleProvider	IBMi_ConcreteRole	Instance & Association
QUME_RoleOfIdentityProvider	IBMi_RoleOfIdentity	Instance & Association
QUME_PrivilegeOfCommonRoleProvider	IBMi_PrivilegeOfCommonRole	Instance & Association
QUME_PrivilegeOfDedicatedRoleProvider	IBMi_PrivilegeOfDedicatedRole	Instance & Association
QUME_OwningRoleElementServiceProvider	IBMi_OwningRoleElement	Instance & Association
QUME_RoleBasedManagementCapabilitiesProvider	IBM_RoleBasedManagementCapabilities	Instance
QUME_RoleAuthorizationProfileProvider	IBMi_RoleAuthorizationProfile	Instance
QUME_RoleBasedAuthorizationServiceProvider	IBM_RoleBasedAuthorizationService	Instance & Method
QUME_ServiceAffectsRoleProvider	IBMi_ServiceAffectsRole	Instance & Association
QUME_ServiceAffectsPrivilegeProvider	IBMi_ServiceAffectsPrivilege	Instance & Association
QUME_ConcreteTemplatePrivilegeProvider	IBMi_ConcreteTemplatePrivilege	Instance & Association

IBM_Identity

This provider will return instances of security principal available on the system when an enumerated list of instances is asked for, or will look up the resource based on the Identity Name provided as the key under the InstanceID property.

Table 35. IBM_Identity

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	Security principal for *ElementName*
string Description	textual description of the object	Security principal for *ElementName*
string ElementName	A user-friendly name for the object.	

Table 35. IBM_Identity (continued)

Property name	Property description	Value or value location
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	

IBMi_UserAccount

This provider will return instances of User Profile available on the system when an enumerated list of instances is asked for, or will look up the resource based on the profile Name provided as the key under the Name property.

Table 36. IBMi_UserAccount

Property name	Property value and data source	Instance mapping rule
string Caption (64)	instance of Systemshort textual description of the object	User *ElementName*
string ElementName	A user-friendly name for the object	
string Name (key) (1024)	The Name property defines the label by which the object is known.	
uint16 OperationalStatus[]	Indicates the current status(es) of the element	2 (OK) / 15 (Dormant)
string StatusDescriptions[]	Strings describing the various OperationalStatus array values	Enabled or Disabled
string SystemCreationClassName (key) (256)	The scoping System's CCN	IBMOS400_ComputerSystem
string SystemName (key) (256)	The scoping System's Name	
string CreationClassName (key) (256)	CreationClassName indicates the name of the class or the subclass used in the creation of an instance	IBMi_UserAccount
string UserID (256)	UserID is the value used by the SecurityService to represent identity	
string Descriptions[] (1024)	The Descriptions property values may contain human-readable descriptions of the object	
string Host[]	Based on RFC1274, the host name of the system(s) for which the account applies. The host name may be a fully-qualified DNS name or it may be an unqualified host name	
string LocalityName[]	This property contains the name of a locality, such as a city, county or other geographic region	
string OrganizationName[]	The name of the organization related to the account	
string OU[]	The name of an organizational unit related to the account	
string UserPassword[]	In the case of an LDAP-derived instance, the UserPassword property may contain an encrypted password used to access the person's resources in a directory	Not retrieved only available to set
string HomeDirectory	The Path to the home directory of the user	
string FullName	The full name or description of the user.	
string GroupId	The Id of the group that the user belongs to	
datetime passwordLastChanged	Days since Jan 1 1970, the password was last changed	

Table 36. IBMi_UserAccount (continued)

Property name	Property value and data source	Instance mapping rule
uint32 daysMustKeepPassword	Days before which the password may not be changed	
uint32 daysMustChangePasswordAfter	Days after which the password must be changed	
datetime passwordExpirationDate	Days since Jan 1 1970, when account will be disabled	
boolean AccountDisable	Is the user account locked? true for YES, false for NO	
string Description	The Description property provides a textual description of the object	
datetime InstallDate	A datetime value that indicates when the object was installed. Lack of a value does not indicate that the object is not installed	
string ObjectClass[]	In the case of an LDAP-derived instance, the ObjectClass property value(s) may be set to the objectClass attribute values	
datetime ProfileChangedDate	The date and time the object was last changed	
boolean GroupMembersIndicator	Whether this user is a group that has members	
string GroupProfileName	Group Profile Name	
datetime PreviousSignon	The date and time the user last signed on	
string PrintDevice	The printer used to print for this user	
string ProfileOwner	This field indicates who is to own objects created by this user	
string SupplementalGroups[]	Supplemental Groups	
string GroupAuthority	The authority the user's group profile has to objects the user creates	
string GroupAuthorityType	The type of authority the user's group profile has to objects the user creates	
uint32 GroupId	The group ID number for the user profile	
uint32 InvalidSignonAttempts	The number of sign-on attempts that were not valid since the last successful sign-on	
string LimitedCapabilities	Whether the user has limited capabilities	
boolean NoPassword	If *NONE is specified for the password in the user profile, this field contains a Y. If not, this field is N	
string AuditingValue	The current user's object auditing value	
uint64 UID	The user ID (UID) number for the user profile	
sint32 PasswordExpirationInterval	The number of days (from 1 through 366) the user's password can remain active before it must be changed	
boolean PasswordExpired	Whether the user's password is set to expire, requiring the user to change the password when signing on	
string SpecialAuthorities[]	The special authorities the user has	
string AuditLevel[]	The action audit values for this user	

Table 36. IBMi_UserAccount (continued)

Property name	Property value and data source	Instance mapping rule
string AccountCode	The accounting code that is associated with this user	
string AssistanceLevel	The user interface that the user will use	
string AttentionProgram	The Attention-key-handling program for this user	
sint32 CCSID	The character code set ID to be used by the system for this user	
string CountryID	The country or region ID used by the system for this user	
string CharacterIdentifierControl	The character identifier control for the user	
string CurrentLibrary	This field contains the name of the user's current library	
boolean DigitialCertificate	Whether there are digital certificates associated with this user	
string DisplaySignon	Whether the sign-on information display is shown when the user signs on	
string InitialMenu	The initial menu for the user	
string InitialProgram	The initial program for the user	
string JobDescription	job description used for jobs that start through subsystem work station entries	
string KeyboardBuffering	This field indicates the keyboard buffering value that is used when a job is initialized for this user	
string LanguageID	The language ID used by the system for this user	
string LimitDeviceSession	Whether the user is limited to one device session	
string Locale	The locale path name that is assigned to the user profile when a job is started	
string LocaleJobAttributes[]	The job attributes that are taken from the user's locale path name	
sint32 MaximumStorage	The maximum amount of auxiliary storage (in kilobytes) that can be assigned to store permanent objects owned by the user	
string MessageQueue	message queue that is used by this user	
string MessageQueueDelivery	How the messages are delivered to the message queue used by the user	
uint32 MessageQueueSeverity	The lowest severity that a message can have and still be delivered to a user in break or notify mode	
string OutputQueue	output queue used by this user	
uint32 SchedulingPriority	The highest scheduling priority the user is allowed to have for each job submitted to the system	
string SortSequenceTable	the sort sequence table used for string comparisons	
string SpecialEnvironment	The special environment the user operates in after signing on	

Table 36. IBMi_UserAccount (continued)

Property name	Property value and data source	Instance mapping rule
uint64 StorageUsed	The amount of auxiliary storage (in kilobytes) occupied by this user's owned objects	
string UserOptions[]	The options for users to customize their environment	
boolean LocalPasswordManagement	Local Password Management	
string BlockPasswordChange	Block password change	
datetime LastUsedDate	The date of the user was last used.	
datetime RestoreDate	The date and time of the user was last restored.	
datetime UserExpirationDate	The date when the user profile expires and is automatically disabled or deleted.	
uint32 UserExpirationInterval	The number of days before the user profile is automatically disabled.	

IBMi_Group

This provider will return instances of Group Profile available on the system when an enumerated list of instances is asked for, or will look up the resource based on the profile Name provided as the key under the Name property. Property list of IBMi_Group is same with that of IBMi_UserAccount. Please refer to table 3 for IBMi_Group properties.

IBMi_UserIdentity

This provider will return association between a security principal and a user account.

Table 37. IBMi_UserIdentity

Property name	Property value and data source	Instance mapping rule
CIM_Identity REF IdentityInfo	Returns all instances of Identity	CIM_Identity REF IdentityInfo CIM_ManagedElement REF ManagedElement.
CIM_ManagedElement REF ManagedElement	Returns instance of User Account	

IBMi_GroupIdentity

This provider will return association between a security principal and a Group profile.

Table 38. IBMi_GroupIdentity

Property name	Property description	Value or value location
CIM_Identity REF IdentityInfo	Returns all instances of Identity	This should be 1 to 1 association between IBM_Identity and IBMi_Group
CIM_ManagedElement REF ManagedElement	Returns instance of Group	

IBMi_UserContactIdentity

This provider will return association between a security principal and a user contact.

Table 39. IBMi_UserContactIdentity

Property name	Property value and data source	Instance mapping rule
CIM_Identity REF IdentityInfo	Returns all instances of Identity	This should be 1 to 1 association between IBM_Identity and IBMi_UserContact
CIM_ManagedElement REF ManagedElement	Returns instance of User Contact	

IBMi_UserContact

This provider will return instances of User Profile contact information available on the system when an enumerated list of instances is asked for, or will look up the resource based on the profile Name provided as the key under the Name property.

Table 40. IBMi_UserContact

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	User *Name*
string Description	The Description property provides a textual description of the object.	
string ElementName	A user-friendly name for the object	*Name*
string CreationClassName(key)	CreationClassName indicates the name of the class or the subclass used in the creation of an instance	IBMi_UserContact
string GivenName	The Given Name property is used for the part of a person's name that is not their surname nor their middle name	
string LocalityName	This property contains the name of a locality, such as a city, county or other geographic region	
string Name(key)	The Name property defines the label by which the object is known	
string Surname	The Surname property specifies the linguistic construct that normally is inherited by an individual from the individual's parent or assumed by marriage, and by which the individual is commonly known	
string TelephoneNumber	The TelephoneNumber property specifies a telephone number of the organization	
string UserID	A User ID property. (User ID)	
string PostalAddress[]	The PostalAddress property values specify the address information required for the physical delivery of postal messages by the postal authority to the person	
boolean AllowSynchronization	Allow Synchronization	
string Building	Building	
string Department	Department	
string Company	Company	
string FaxNumber	Fax Number	
string FirstName	First Name	
string FullName	Full Name	
string JobTitle	Job Title	
string MailServiceLevel	Mail Service Level	
string MiddleName	Middle Name	
string Office	Office	
string PreferredAddress	Preferred Address	
string SMTPDomain	SMTP Domain	
string PreferredAddressID	Preferred Address ID	
string PreferredAddressType	Preferred Address Type	

Table 40. IBMi_UserContact (continued)

Property name	Property description	Value or value location
string SMTPRoute	SMTP Route	
string SMTPUserID	SMTP User ID	
string SystemName	System Name	
string SystemGroup	System Group	
string UserAddress	User Address	

IBMi_MemberOfGroup

This provider will return association between a security principal and a user Group.

Table 41. IBMi_MemberOfGroup

Property name	Property description	Value or value location
CIM_Collection REF Colection	Returns all instances of Group	This should be 1 to N association between IBMi_Group and IBM_Identity. Find all members in one User Group
CIM_ManagedElement REF Member	Returns instance of Identity	

IBM_AccountManagementCapabilities

This provider describes the capabilities supported for managing Accounts service.

Table 42. IBM_AccountManagementCapabilities

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Account Management Capabilities
string Description	The Description property provides a textual description of the object.	Capabilities supported for managing Accounts associated with an instance of AccountManagementService
string ElementName	A user-friendly name for the object	IBM i Account Management Capabilities
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	IBM i Account Management Capabilities
boolean ElementNameEditSupported	Boolean indicating whether the ElementName can be modified.	FALSE
uint16 RequestedStatesSupported[]	RequestedStatesSupported indicates the possible states that can be requested when using the method RequestStateChange on the EnabledLogicalElement	2(Enabled), 3(Disabled)
uint16 OperationsSupported[]	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	2(Create), 3(Modify), 4(Delete)

IBM_AccountManagementService

This provider describes the services supported for managing Accounts.

Table 43. IBM_AccountManagementService

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Account Management Service
string Description	The Description property provides a textual description of the object.	Services supported for managing Accounts

Table 43. IBM_AccountManagementService (continued)

Property name	**Property value and data source**	**Instance mapping rule**
string ElementName	A user-friendly name for the object	Account Management Service
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element	2(Enabled)
uint16 RequestedState = 12	RequestedState is an integer enumeration that indicates the last requested or desired state for the element	5(No change)
uint16 EnabledState = 5	EnabledState is an integer enumeration that indicates the enabled and disabled states of an element	2(Enabled)
string CreationClassName(key)	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance	IBM_AccountManagementService
string Name(key)	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed	Account Management Service
boolean Started	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE)	TRUE
string StartMode	The use of this element is deprecated in lieu of the EnabledDefault property that is inherited from EnabledLogicalElement	Automatic
string SystemCreationClassName(key)	The CreationClassName of the scoping System	IBMOS400_ComputerSystem
string SystemName(key)	The Name of the scoping System	

IIBM_AccountOnSystem

This provider describes all CIM_Account instances on the system.

Table 44. IBM_AccountOnSystem

Property name	**Property value and data source**	**Instance mapping rule**
CIM_Account REF PartComponent	Returns instance of User account	This should be 1 to N association between IBM_ComputerSystem and IBMi_UserAccount
CIM_System REF GroupComponent	Returns instance of Computer System	

IBMi_AccountElementCapabilities

This provider represents the association between CIM_AccountManagementService and CIM_AccountManagementCapabilities.

Table 45. IBMi_AccountElementCapabilities

Property name	**Property value and data source**	**Instance mapping rule**
CIM_Capabilities REF Capabilities	Returns instance of User account capabilities	This should be 1 to 1 association between IBM_AccountManagementCapabilities and IBM_AccountManagementService
CIM_ManagedElement REF ManagedElement	Returns instance of Account Management Service	

IBMi_HostedAccountService

This provider represents the association between CIM_AccountManagementService and IBM_ComputerSystem.

Table 46. IBMi_HostedAccountService

Property name	Property value and data source	Instance mapping rule
CIM_System REF Antecedent	Returns instance of Computer System	This should be 1 to N association between IBM_ComputerSystem and IBM_AccountManagementService
CIM_Service REF Dependent	Returns instance of Account management Service	

IBMi_OwningAccountElemen

This provider represents all User Groups on a computer system.

Table 47. IBMi_OwningAccountElement

Property name	Property description	Value or value location
CIM_Collection REF OwnedElement	Returns instance of User Group	This should be 1 to N association between IBM_ComputerSystem and IBMi_Group
CIM_ManagedElement REF OwningElement	Returns instance of Computer System	

IBMi_ServiceAffectsAccount

This provider represents the association between CIM_AccountManagementService and IBM_Identity.

Table 48. IBMi_ServiceAffectsAccount

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF AffectedElement	Returns instance of User Identity	This should be 1 to N association between IBM_Identity and IBM_AccountManagementService
CIM_Service REF AffectingElement	Returns instance of Account management Service	

IBMi_AccountManagementRegisteredProfile

This provider will return instances of registered profile for Account management.

Table 49. IBMi_AccountManagementRegisteredProfile

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Account Management Service
string Description	The Description property provides a textual description of the object.	Services supported for managing Accounts
string ElementName	A user-friendly name for the object	Account Management Service
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Account Management Registered Profile
string RegisteredName	The name of this registered profile	Distributed Management Task Force (DMTF): Simple Identity Management Profile
uint16 RegisteredOrganization	The organization that defines this profile	2
string RegisteredVersion	The version of this profile.	1.0.0
uint16 AdvertiseTypes[]	This property signifies the advertisement for the profile information.	3 (Service Location Protocol (SLP))

IBMi_AccountElementConformsToProfile

This provider will return association between Simple Identity Profile and Account management Service.

Table 50. IBMi_AccountElementConformsToProfile		
Property name	**Property value and data source**	**Instance mapping rule**
CIM_RegisteredProfile REF ConformantStandard	instance of Simple Identity Profile	This should be 1 to 1 association between IBMi_AccountManagementRegisteredProfile and IBM_AccountManagementService
CIM_ManagedElement REF ManagedElement	Returns instance of Account management Service.	

IBMi_ServicePrivilege

This provider represents the association between CIM_RoleBasedAuthorizationService and IBM_Privilege.

Table 51. IBMi_ServicePrivilege		
Property name	**Property value and data source**	**Instance mapping rule**
CIM_ManagedElement REF Antecedent	Returns instance of role based authorization service	This should be 1 to N association between IBM_RoleBasedAuthorizationService and IBM_Privilege
CIM_ManagedElement REF Dependent	Returns instance of privileges	

IBMi_RoleElementCapabilities

This provider represents the association between CIM_AccountManagementService and CIM_AccountManagementCapabilities.

Table 52. IBMi_RoleElementCapabilities		
Property name	**Property value and data source**	**Instance mapping rule**
CIM_Capabilities REF Capabilities	Returns instance of Role Based Management capabilities	This should be 1 to 1 association between IBM_RoleBasedManagementCapabilities and IBM_RoleBasedAuthorizationService
CIM_ManagedElement REF ManagedElement	Returns instance of Role based authorization service	

IBMi_HostedRoleService

This provider represents the association between CIM_AccountManagementService and IBM_ComputerSystem.

Table 53. IBMi_HostedRoleService		
Property name	**Property value and data source**	**Instance mapping rule**
CIM_System REF Antecedent	Returns instance of Computer System	This should be 1 to N association between IBM_ComputerSystem and IBM_RoleBasedAuthorizationService
CIM_Service REF Dependent	Returns instance of Role Based Authorization Service	

IBMi_IBMi_CommonRole

IBMi_CommonRole model IBM i user class of user profile. There are totally five kinds of user classes: “USER”, “SYSOPR”, “PGMR”, “SECADM”, “SECOFR”.

Table 54. IBMi_CommonRole		
Property name	**Property value and data source**	**Instance mapping rule**
string Caption	The Caption property is a short textual description (one- line string) of the object	User Class *Name*
string Description	The Description property provides a textual description of the object.	User Class *Name*
string ElementName	A user-friendly name for the object	*Name*

Table 54. IBMi_CommonRole (continued)

Property name	Property value and data source	Instance mapping rule
string CommonName	A Common Name is a (possibly ambiguous) name by which the role is commonly known in some limited scope (such as an organization) and conforms to the naming conventions of the country or culture with which it is associated	hostname : *Name*
string CreationClassName(key)	CreationClassName indicates the name of the class or the subclass used in the creation of an instance	IBMi_CommonRole
string Name(key)	The Name property defines the label by which the object is known	*USER, *SYSOPR, *PGMR, *SECADM, *SECOFR

IBMi_DedicatedRole

On IBM i, special authority of a user profile is not decided by role, actually, it is decided by profile name. When special authority granted to a user profile is different from the ones system standard role has defined, it becomes a user specific role. To find special authority of a user profile, we need associate to a user customized role but common role.

Table 55. IBMi_DedicatedRole

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	User Class *Name*
string Description	The Description property provides a textual description of the object.	User Class *Name*
string ElementName	A user-friendly name for the object	*Name*
string CommonName	A Common Name is a (possibly ambiguous) name by which the role is commonly known in some limited scope (such as an organization) and conforms to the naming conventions of the country or culture with which it is associated	*Name*
string CreationClassName(key)	CreationClassName indicates the name of the class or the subclass used in the creation of an instance	IBMi_DedicatedRole
string Name(key)	The Name property defines the label by which the object is known	CIM_Account.Name_CIM_Role.Name

IBM_Privilege

IBM_Privilege models IBM i special authorities of a user profile.

Table 56. IBM_Privilege

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Special authority *InstanceID*
string Description	The Description property provides a textual description of the object.	*InstanceID*
string ElementName	A user-friendly name for the object	*InstanceID*
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	*ALLOBJ, *AUDIT, *IOSYSCFG, *JOBCTL, *SAVSYS, *SECADM, *SERVICE, *SPLCTL
boolean PrivilegeGranted	Boolean indicating whether the Privilege is granted (TRUE) or denied (FALSE)	TRUE

IBMi_RoleOfIdentity

This provider will return association between a security principal and a System defined Role.

Table 57. IBMi_RoleOfIdentity

Property name	Property value and data source	Instance mapping rule
CIM_Collection REF Collection	Returns all instances of Role	This should be 1 to 1 association between IBMi_CommonRole and IBM_Identity.
CIM_ManagedElement REF Member	Returns instance of Identity	

IBMi_ConcreteRole

This provider will return association between a security principal and a dedicated Role.

Table 58. IBMi_ConcreteRole

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF Antecedent	Returns all instances of Role	This should be 1 to 1 association between IBMi_CommonRole and IBM_Identity.
CIM_ManagedElement REF Dependent	Returns instance of Identity	

IBMi_PrivilegeOfCommonRole

This provider will return association between a privilege and a system defined Role.

Table 59. IBMi_PrivilegeOfCommonRole

Property name	Property value and data source	Instance mapping rule
CIM_Collection REF Collection	Returns all instances of Role	This should be 1 to N association between IBMi_CommonRole and IBM_Privilege. Find all privileges of one Role
CIM_ManagedElement REF Member	Returns instance of Privilege	

IBMi_PrivilegeOfDedicatedRole

This provider will return association between a privilege and a user specific Role.

Table 60. IBMi_PrivilegeOfDedicatedRole

Property name	Property value and data source	Instance mapping rule
CIM_Collection REF Collection	Returns all instances of custom Role	This should be 1 to N association between IBMi_DedicatedRole and IBM_Privilege. Find all privileges of one Role.
CIM_ManagedElement REF Member	Returns instance of Privilege	

IBMi_OwningRoleElement

This provider represents all Roles on a computer system.

Table 61. IBMi_OwningRoleElement

Property name	Property value and data source	Instance mapping rule
CIM_Collection REF OwnedElement	Returns instance of Role	This should be 1 to N association between IBM_ComputerSystem and CIM_Role
CIM_ManagedElement REF OwningElement	Returns instance of Computer System	

IBM_RoleBasedManagementCapabilities

This provider describes the capabilities supported for Role Based Authorization service.

Table 62. IBM_RoleBasedManagementCapabilities

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Role Based Management Capabilities
string Description	The Description property provides a textual description of the object.	Capabilities supported for managing Role
string ElementName	A user-friendly name for the object	IBM i Role Based Management Capabilities
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	IBM i Role Based Management Capabilities
boolean SharedPrivilegeSupported	Boolean indicating whether the ElementName can be modified	TRUE
uint16 ActivitiesSupported[]	This string expresses the restrictions on ElementName	4,5,6,7
string ActivityQualifiersSupported[]	Maximum supported ElementName length	Detect, Read, Write, Execute
uint16 QualifierFormatsSupported[]	RequestedStatesSupported indicates the possible states that can be requested when using the method RequestStateChange on the EnabledLogicalElement	4(Method)
uint16 SupportedMethods []	StateAwareness indicates support for modeling the state of the associated instance of CIM_EnabledLogicalElement	1(ShowAccess),2(AssignAccess),3(RevokeAccess)

IBMi_RoleAuthorizationProfile

This provider will return instances of registered profile for Role Authorization.

Table 63. IBMi_RoleAuthorizationProfile

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Role Authorization Registered Profile
string Description	The Description property provides a textual description of the object.	Role Authorization Registered Profile
string ElementName	A user-friendly name for the object	Role Authorization Registered Profile
string InstanceID(key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Role Authorization Registered Profile
string RegisteredName	The name of this registered profile	Distributed Management Task Force (DMTF): Role Based Authorization Profile
uint16 RegisteredOrganization	The organization that defines this profile	2
string RegisteredVersion	The version of this profile	1.0.0
uint16 AdvertiseTypes[]	This property signifies the advertisement for the profile information.	3 (Service Location Protocol (SLP))

IBM_RoleBasedAuthorizationService

This provider describes the services supported for managing roles.

Table 64. IBM_RoleBasedAuthorizationService

Property name	Property value and data source	Instance mapping rule
string Caption (64)	short textual description of the object	Role Based Authorization Service
string Description	The Description property provides a textual description of the object.	Role Based Authorization Service

Table 64. IBM_RoleBasedAuthorizationService (continued)

Property name	Property value and data source	Instance mapping rule
string ElementName	A user-friendly name for the object	Role Based Authorization Service
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element	2(Enabled)
uint16 RequestedState = 12	RequestedState is an integer enumeration that indicates the last requested or desired state for the element	5(No change)
uint16 EnabledState = 5	EnabledState is an integer enumeration that indicates the enabled and disabled states of an element	2(Enabled)
string CreationClassName(key)CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance	The version of this profile	IBM_RoleBasedAuthorizationService
string Name(key)	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed	Role Based Authorization Service
boolean Started	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	TRUE
string StartMode	The use of this element is deprecated in lieu of the EnabledDefault property that is inherited from EnabledLogicalElement	Automatic
string SystemCreationClassName(key)	The CreationClassName of the scoping System	IBMOS400_ComputerSystem
string SystemName(key)	The Name of the scoping System	

IBMi_ServiceAffectsRole

This provider represents the association between CIM_RoleBasedAuthorizationService and CIM_Role.

Table 65. IBMi_ServiceAffectsRole

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF AffectedElement	Returns instance of role	There should be 1 to N association between CIM_Role and IBM_RoleBasedAuthorizationService
CIM_Service REF AffectingElement	Returns instance of Role Based Authorization Service	

IBMi_ServiceAffectsPrivilege

This provider represents the association between CIM_RoleBasedAuthorizationService and IBM_Privilege.

Table 66. IBMi_ServiceAffectsPrivilege

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF AffectedElement	Returns instance of privilege	There should be 1 to N association between IBM_Privilege and IBM_RoleBasedAuthorizationService
CIM_Service REF AffectingElement	Returns instance of Role Based Authorization Service	

IBMi_ConcreteTemplatePrivilege

This provider will return association between Role based authorization service and template privilege.

Table 67. IBMi_ConcreteTemplatePrivilege

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF Antecedent	Returns all instances of Role based authorization service	There should be 1 to N association between IBM_RoleBasedAuthorizationService and IBM_Privilege
CIM_ManagedElement REF Dependent	Returns instance of template privilege	

System inventory providers

These providers support to view or change system values, system configuration, network and service attributes. This section includes four categories of system related providers. Properties changes in these providers may affect the whole system for they are globe control label.

Supported providers

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

Table 68. Supported Providers

Provider name	Implements CIM class	Provider type	Category
QUME_AuditingSettingDataProvider	IBMi_AuditingSettingData	Instance	System value
QUME_DateTimeSettingDataProvider	IBMi_DateTimeSettingData	Instance	System value
QUME_DevicesSettingDataProvider	IBMi_DevicesSettingData	Instance	System value
QUME_InternationalSettingDataProvider	IBMi_InternationalSettingData	Instance	System value
QUME_JobSettingDataProvider	IBMi_JobSettingData	Instance	System value
QUME_MessageServiceSettingDataProvider	IBMi_MessageServiceSettingData	Instance	System value
QUME_PasswordSettingDataProvider	IBMi_PasswordSettingData	Instance	System value
QUME_PerformanceSettingDataProvider	IBMi_PerformanceSettingData	Instance	System value
QUME_PowerControlSettingDataProvider	IBMi_PowerControlSettingData	Instance	System value
QUME_PrintingSettingDataProvider	IBMi_PrintingSettingData	Instance	System value
QUME_RestartSettingDataProvider	IBMi_RestartSettingData	Instance	System value
QUME_SaveRestoreSettingDataProvider	IBMi_SaveRestoreSettingData	Instance	System value
QUME_SecuritySettingDataProvider	IBMi_SecuritySettingData	Instance	System value
QUME_SignonSettingDataProvider	IBMi_SignonSettingData	Instance	System value
QUME_StorageSettingDataProvider	IBMi_StorageSettingData	Instance	System value
QUME_SystemUserSettingDataProvider	IBMi_SystemUserSettingData	Instance	System value
QUME_LibraryListSettingDataProvider	IBMi_LibraryListSettingData	Instance	System value
QUME_ComputerSystemProvider	IBMOS400_ComputerSystem	Instance	System configuration
QUME_NetworkSettingDataProvider	IBMi_NetworkSettingData	Instance	Network attributes
QUME_ServiceSettingDataProvider	IBMi_ServiceSettingData	Instance	Service attributes
QUME_JobQueueProvider	IBM_JobQueue	Instance	System configuration

Table 68. Supported Providers (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_SubsystemProvider	IBM_Subsystem	Instance	System configuration

IBMi_AuditingSettingData

This provider will return auditing system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 69. IBMi_AuditingSettingData

Property name	Property description	Value or value location
string Caption	The Caption property is a short textual description (one- line string) of the object	Auditing System Value
string Description	The Description property provides a textual description of the object	Auditing System Value
string ElementName	A user-friendly name for the object.	Auditing System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Auditing
string AuditingControl[]	Auditing control	
string SecurityAuditingLevel []	Security auditing level	
string ActionToAuditingEnd	Auditing end action	
sint32 ForceAuditingData	Force auditing data	
string CreateObjectAuditing	Create object auditing	
string AuditingLevelExtension[]	Security auditing level extension(Security)	

IBMi_DateTimeSettingData

This provider will return Date and Time system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 70. IBMi_DateTimeSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Date and Time System Value association
string Description	textual description of the object	Date and Time System Value
string ElementName	A user-friendly name for the object	Date and Time System Value

Table 70. IBMi_DateTimeSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	DateTime
string SystemDate	System date	
uint32 LeapYearAdjustment	Leap year adjustment	
string TimeOfDay	Time of day	
string TimeAdjustment	Time adjustment	
string TimeZone	Time zone	
string Century	Century(DateTime)	
string SystemDatetime[]	System date and time(Datetime)	
string Day	Day(DateTime)	
string DayOfWeek	Day of week(DateTime)	
string Hour	Hour of the day(dateTime)	
string Minute	Minute of the hour(DateTime)	
string Month	Month of the year(Datetime)	
string Second	Second of the minute(DateTime)	
string UTCOffset	Coordinated universal time offset(DateTime)	
string Year	Year(DateTime)	

IBMi_DevicesSettingData

This provider will return device system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 71. IBMi_DevicesSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Devices System ValueDevices System Value
string Description	textual description of the object	Devices System Value
string ElementName	A user-friendly name for the object	Devices System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Devices

Table 71. IBMi_DevicesSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string AutoconfigureDevices	Autoconfigure devices	
string DeviceNamingConventions	Device naming conventions	
string AutoConfigureRemoteControllers	Autoconfigure of remote controllers	
sint32 AutoConfigureVirtualDevices	Autoconfigure virtual devices	
string ActionToDeviceIOError	Device I/O error action	

IBMi_InternationalSettingData

This provider will return International system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 72. IBMi_InternationalSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	International System Value
string Description	textual description of the object	International System Value
string ElementName	A user-friendly name for the object	International System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	International
string DateFormat	Date format	
string DateSeparator	Date separator	
string TimeSeparator	Time separator	
string DecimalFormat	Decimal format	
string CurrencySymbol	Currency symbol	
string LanguageID	Language identifier	
string CountryRegionID	Country or region identifier	
string KeyboardType	Keyboard language character set	
uint32 CodedCharSetID	Coded character set identifier	
string GraphicCharacterSetAndCodePage	Graphic character set and code page	
string CharacterIDControl	Character identifier control	
string SortSequence	Sort sequence	
string DBCSSystemIndicator	DBCS version installed indicator	

Table 72. IBMi_InternationalSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string CodedFontName	Double byte code font	
real32 CodedFontSize	Coded font size	
string LocalePathName	Locale path name	
string SetJobAttributesFromLocale[]	Set job attributes from locale	

IBMi_JobSettingData

This provider will return job system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 73. IBMi_JobSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Job System Value
string Description	textual description of the object	Job System Value
string ElementName	A user-friendly name for the object	Job System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Job
uint32 MaxJobsNumber	Maximum number of jobs	
uint32 InitialActiveJobsNumber	Initial number of active jobs	
uint32 InitialTotalJobsNumber	Initial total number of jobs	
uint32 AdditionalActiveJobsNumber	Additional number of active jobs	
uint32 Additional number of total jobsAdditionaTotalJobslNumber	Currency symbol	
string JobLogOutputManner	Job log output	
uint32 MaxJobLogSize	Maximum size of job message queue	
string ActionToJobMessageQueueFull	Job message queue full action	
String InactiveJobTimeoutInterval	Inactive job time-out	
String ActionToInactiveJobTimeout	Inactive job message queue	
String DisconnectJobTimeoutInterval	Time interval before disconnected jobs end	
String ActionToNotThreadsafe	Multithreaded job action	

Table 73. IBMi_JobSettingData (continued)

Property name	Property value and data source	Instance mapping rule
String IsSpooledFileDetached	Spooled file action	
uint32 SpoolingControlBlockInitialSize	Spooling control block initial size	
uint32 MaxSpoolFilesNumber	Maximum spooled files	
uint32 TimeLimitOfEndingJob	Time limit during immediate ending of a job	
uint32 IsAllowJobInterrupted	Allow jobs to be interrupted	
uint32 JobMessageQueueInitialSize	Job message queue initial size(Allocation)	
uint32 MaxInitialSizeOfJobMessageQueue	Job message queue maximum initial size(Allocation)	

IBMi_MessageServiceSettingData

This provider will return Message Service system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 74. IBMi_MessageServiceSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Message and Service System Value
string Description	textual description of the object	Message and Service System Value
string ElementName	A user-friendly name for the object	Message and Service System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	MessageService
uint32 MaxHistoryLogRecords	Maximum history log records	
string IsDisplayStatusMessages	Display status messages	
string MessageQueue	Configuration message queue	
string AccountingLevel[]	Accounting level	
string IsProblemLogFiltered	Problem log filter	
uint32 MinRetentionOfProblemLog	Problem log hold interval	
string IsSoftwareProblemLogged	Software error logging	
string ServiceDumpControl	Service dump control	
string IsAnalyzedRemotely	Remote service attribute	

IBMi_PasswordSettingData

This provider will return Password system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 75. IBMi_PasswordSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Password System Value
string Description	textual description of the object	Password System Value
string ElementName	A user-friendly name for the object	Password System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Password
uint32 PasswordLevel	Password level	
uint32 MinPasswordLength	Minimum password length	
uint32 MaxPasswordLength	Maximum password length	
string IsDigitRequiredInPassword	Require digit in password	
string IsAdjacentDigitsAllowed	Limit adjacent digits in password	
string CharatersRestricted	Limit characters in password	
string IsRepeatingCharactersAllowed	Limit repeating characters in password	
string PasswordReuseCycle	Duplicate password control	
string IsNewCharacterEachPositionRequired	Limit password character positions	
string PasswordExpirationInterval	Password expiration interval	
string PasswordValidationProgram	Password validation program(security)	
string PasswordRules[]	Password rules(Security)	
string BlockPasswordChange	Block password change(Security)	
uint32 PasswordExpirationWarning	Password expiration warning(Security)	

IBMi_PerformanceSettingData

This provider will return Performance system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 76. IBMi_PasswordSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Performance System Value
string Description	textual description of the object	Performance System Value
string ElementName	A user-friendly name for the object	Performance System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Performance
string DynamicPriorityAdjustment	Dynamic priority adjustment	
string DynamicPriorityScheduler	Dynamic priority scheduler	
string PerformanceAdjustment	Performance adjustment	
sint32 MaxActivityLevel	Maximum activity level of system	

Table 76. IBMi_PasswordSettingData (continued)

Property name	Property value and data source	Instance mapping rule
uint32 MachineMemoryPoolSize	Machine storage pool size	
uint32 MinBaseMemoryPoolSize	Base storage pool minimum size	
uint32 MaxActivityLevelOfBaseMemoryPool	Base storage pool activity level	
string ActionToTimeSliceEnd	Time slice end pool	
string CommunicationRecoveryLimit	Communications recovery limits	
string CommunicationArbitersSize	Communication arbiters	
string DisplayStationPassThroughServerJobsSize	Pass-through servers	
string ParallelProcess	Parallel processing degree	
string DatabaseQueryTimeLimit	Query processing time limit	
string DatabaseFileStatisticsCollectionType	Database file statistics collection	
string LockLibrary	Library locking level	
string ThreadAffinityLevel[]	Thread resources affinity	
string DynamicAdjustmentToThreadAffinity	Thread resources adjustment	

IBMi_PowerControlSettingData

This provider will return instances of all Storage Controller available on the system when an enumerated list of instances is asked for, or will look up the resource based on the Logical Resource Name provided as the key under the DeviceID property.

Table 77. IBMi_PowerControlSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one-line string) of the object	Power® Control System Value
string Description	textual description of the object	Power Control System Value
string ElementName	A user-friendly name for the object	Power Control System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Power Control
string ActionToPowerFailure	Uninterruptible power supply delay time	
string UPSMessageQueueAndLibrary	Uninterruptible power supply message queue	

IBMi_PrintingSettingData

This provider will return Printing system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 78. IBMi_PrintingSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Printing System Value
string Description	textual description of the object	Printing System Value
string ElementName	A user-friendly name for the object	Printing System Value

Table 78. IBMi_PrintingSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Printing
string DefaultPrinter	Printer device description	
string PrintFormat	Print header and/or border information	
string PrintPageFooter	Print text	

IBMi_RestartSettingData

This provider will return Restart system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 79. IBMi_RestartSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Restart System Value
string Description	textual description of the object	Restart System Value
string ElementName	A user-friendly name for the object	Restart System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Restart
string RestartType	Type of IPL to perform	
string AutoRestartOnPowerRestore	Automatic IPL after power restored	
string RemotePowerOn	Remote power on and IPL	
string ScheduledRestart	Date and time to automatically IPL	
uint32 ShutdownTimeLimit	Maximum time for PWRDWNSYS *IMMED	
string StartupProgram	Startup program	
string ControllingSubsytemAndLibrary	Controlling subsystem	
string ActionToConsoleProblem	IPL action with console problem	
string WaitforDatabaseRecovery	Database recovery wait indicator	
string PreviousEndingStatus	Previous end of system indicator(this property can not be set)	
string PreviousRestartType	IPL status indicator(this property can not be set)	
string PreviousPrinterStart	Start print writers at IPL(this property can not be set)	

IBMi_SaveRestoreSettingData

This provider will return Save and Restore system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 80. IBMi_SaveRestoreSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Save and Restore System Value
string Description	textual description of the object	Save and Restore System Value
string ElementName	A user-friendly name for the object	Save and Restore System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Save and Restore
string SignatureVertificationPolicy	Verify object on restore	
string ConversionLevel	Force conversion on restore	
string ObjectRestoreOption[]	Allow object restore option	
string SaveAccessPath	Save access paths	

IBMi_SecuritySettingData

This provider will return Security system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 81. IBMi_SecuritySettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Security System Value
string Description	textual description of the object	Security System Value
string ElementName	A user-friendly name for the object	Security System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Security
string SecurityLevel	System security level	
string RetainServerSecurityData	Retain server security data	
string AdoptedAuthorityUser	Use adopted authority	
string DefaultCreateAuthority	Create default public authority	
string NotAuditingObjects[]	Allow user domain objects in libraries	

Table 81. IBMi_SecuritySettingData (continued)

Property name	Property value and data source	Instance mapping rule
string ScanObjectAtExitProgram[]	Scan file systems	
string ScanControlOptions []	Scan file systems control	
string SharedMemoryControl	Shared memory control	
string SSLCipherSpecificationList[]	V6R1: Secure sockets layer cipher specification list(security)	
string SSLCipherControl	V6R1: Secure sockets layer cipher control(Security)	
string SSLProtocols[]	V6R1: Secure sockets layer protocols(Security)	

IBMi_SignonSettingData

This provider will return Sign on system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 82. IBMi_SignonSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Signon System Value
string Description	textual description of the object	Signon System Value
string ElementName	A user-friendly name for the object	Signon System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Signon
string MaxInvalidSignonAttempts	Maximum sign-on attempts allowed	
string ActionToMaxInvalidSignonAttempts	Action to take for failed signon attempts	
string SignonInformation	Sign-on display information control	
string SpecificDeviceAccessForPrivilegedUser	Limit security officer device access	
string DeviceSessionLimit	Limit device sessions	
string RemoteSignonControl	Remote sign-on control	

IBMi_StorageSettingData

This provider will return Storage system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 83. IBMi_SignonSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Storage System Value
string Description	textual description of the object	Storage System Value
string ElementName	A user-friendly name for the object	Storage System Value

Table 83. IBMi_SignonSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Storage
real32 StorageUtilizationLimit	Auxiliary storage lower limit	
string ActionToMaxStorageUtilization	Auxiliary storage lower limit action	
string RetentionOfSpoolStorage	Reclaim spool storage	
uint32 AdditionalStorage	Spooling control block additional storage(Allocation system values)	
uint32 InitialSpoolSize	Spooling control block initial size(Allocation)	

IBMi_SystemUserSettingData

This provider will return System and User system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 84. IBMi_SystemUserSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	System and User System Value
string Description	textual description of the object	System and User System Value
string ElementName	A user-friendly name for the object	System and User System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	System and User
string SystemModelNumber	System model number(this property can not be set)	
string SystemSerialNumber	System serial number(this property can not be set)	
string ProcessorFeature	Processor feature(this property can not be set)	
string ConsoleName	Console name(this property can not be set)	
string AssistanceLevel	User assistance level	
string AttentionProgram	Attention program	
string SpecialEnvironment	Special environment	
string KeyboardBufferType	Type ahead and/or attention key option	

Table 84. IBMi_SystemUserSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string AutoReportingProblem	Automatic system disabled reporting(System control system values.)	
string BookSearchPath[]	Book and bookshelf search path(System control system values)	
string CharacterIDControl	Character identifier control(System control system values)	
string ProcessorMultitasking	Processor multi tasking(System control system values.)	
uint32 ServerAuthenticationInterval	Server authentication interval(system control)	

IBMi_LibraryListSettingData

This provider will return Library list system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 85. IBMi_LibraryListSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Library List System Value
string Description	textual description of the object	Library List System Value
string ElementName	A user-friendly name for the object	Library List System Value
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Library List
string SystemLibraryList[]	System part of the library list	
string UserLibraryList[]	User part of the library list	

IBMOS400_ComputerSystem

This provider will return instances of Computer System.

Table 86. IBMOS400_ComputerSystem

Property name	Property value and data source	Instance mapping rule
string Description	A textual description of the object	
string ElementName	A user-friendly name for the object	*DeviceID*
uint16 HealthState	The current health of the element	
string Name(1024)	The label by which the object is known	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element	
string StatusDescriptions[]	The various OperationalStatus array values	

Table 86. IBMOS400_ComputerSystem (continued)

Property name	Property value and data source	Instance mapping rule
uint16 EnabledDefault = 2	An enumerated value indicating 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	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system	
string PointOfOrigin	Point of Origin	
string ProcessorCUoD	Processor CUoD Capacity Card Info	
string MemoryCUoD	Memory CUoD Capacity Card Info	
string MachineTypeModel	Machine type-model	
string MachineSerial	Machine Serial Number	
string SystemFeatureCode	System feature code	
string ProcessorFeatureCode	Processor Feature Code	
string FixLevel	Fix Level	
uint32 StorageCapacity	The storage capacity of the system auxiliary storage pool (ASP1). This value is in millions (M) of bytes	
real32 StorageUsed	The percentage (in ten thousandths) of the system storage pool currently in use	
uint64 InstalledMemory	The amount of main storage, in kilobytes, in the pool	
string PrimaryPartitionID	the identifier of the primary partition	
string NumberOfPartitions	the number of partitions that are active on the current physical machine IPL	
uint16 ActiveProcessors	the number of virtual processors currently active in the partition	
uint16 TotalProcessors	number of processors installed on the physical machine	

Table 86. IBMOS400_ComputerSystem (continued)		
Property name	**Property value and data source**	**Instance mapping rule**
string CustomerNumber	Customer Number	

IBMi_NetworkSettingData

This provider will return network system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 87. IBMi_NetworkSettingData		
Property name	**Property value and data source**	**Instance mapping rule**
string Caption	The Caption property is a short textual description (one- line string) of the object	Network Attributes
string Description	textual description of the object	Network Attributes
string ElementName	A user-friendly name for the object	*InstanceID*
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Network Attributes
string AlertBackupFocalPoint	Alert backup focal point	
string AlertController	Alert controller description	
string IsAlertFocalPoint	Alert default focal point	
string AlertFilter	Alert filter	
sint32 AlertHoldCount	Alert hold count	
string AlertLogStatus	Alert logging status	
string IsAlertPrimaryFocalPoint	Alert primary focal point	
string RequestingAlertFocalPoint	Alert focal point to request	
string AlertStatus	Alert status	
string IsAllowAddToCluster	Allow add to cluster	
string IsAnyNetSupport	Allow AnyNet® support	
string IsHPRTowerSupport	Allow HPR transport tower	
string IsVirtualAPPNSupport	Allow APPN virtual support	
uint32 AutocreateAPPCDeviceLimit	Virtual ctl autocreate dev	
string DDMAndDRDARequestAccess	DDM/DRDA request access	
string DefaultISDNConnectionList	Default ISDN connection list	
string DefaultMode	Specifies the default mode for APPC and APPN locations	

Table 87. IBMi_NetworkSettingData (continued)

Property name	Property value and data source	Instance mapping rule
string ISDNNetworkType	Default ISDN network type	
sint32 DataCompressionOption	Data compression	
sint32 IntermediateDataCompressionLevel	Intermediate data compression	
string HPRPathSwitchTimers	HPR path switch timers	
string JobAction	Specifies the action taken for input streams received through the SNADS network by the system	
string LocalControlPointName	Local control point name	
string DefaultLocalLocationName	Default local location name	
string LocalNetworkID	Local network ID	
uint32 MaxIntermediateSessionsNumber	Maximum intermediate sessions	
uint32 MaxHopCount	Maximum hop count	
string ModemCountryRegionID	Modem country or region ID	
string MessageQueueBySNADS	Message queue	
string ServerNetworkID[]	Network node servers	
string APPNNodeType	Specifies the APPN node type	
string NetworkServerDomain	Network server domain	
string OutputQueueName	Specifies the name and library of the output queue to which spooled files received through the SNADS network are sent for users whose output queue is not available	
string PendingSystemName	Pending system name	
string ClientRequestAccess	Client request access	
uint32 RouteAdditionResistance	Route addition resistance	
string SystemName	System name	

IBMi_ServiceSettingData

This provider will return service system values available on the system when an enumerated list of instances is asked for, or will look up the resource based on the value name provided as the key under the InstanceID property.

Table 88. IBMi_ServiceSettingData

Property name	Property value and data source	Instance mapping rule
string Caption	The Caption property is a short textual description (one- line string) of the object	Service Attributes
string Description	textual description of the object	Service Attributes
string ElementName	A user-friendly name for the object	*InstanceID*
string InstanceID	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	Service Attributes
string AutoProblemAnalysis	Automatic problem analysis	
string AutoProblemReporting	Automatic problem reporting	
string ReportProblemServiceProvider	Service provider to report problem	
string PTFInstallType	PTF install type	
string CriticalMessageRecipients[]	Critical message recipients	
string SendDataPackets	Send data packets	
string CopyPTFs	Copy PTFs	
string SystemdisableRepportConnectionNumber	System-disabled reporting connection number	
string SystemDisableCallbackConnectionNumber	System-disabled call-back connection number	
string ServiceProviderConnectionNumber	Service provider connection number	

IBM_JobQueue

This provider looks up a resource based on the logical resource name that is provided as the key under the Name property, and returns instances of all job queues that are available on the system.

Table 89. IBM_JobQueue

Property name	Property Description	Value or Value Location
string SystemCreationClassName (key)	The CreationClassName of the scoping System.	IBMOS400_ComputerSystem
string SystemName (key)	The Name of the scoping System.	System name
string CreationClassName (key)	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	IBM_JobQueue
string Name (key)	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	Job queue library name/Job queue name
string Caption	A short textual description of the object.	Job Queue
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	Job queue library name/Job queue name
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element.	5 (No Change)
uint16 EnabledDefault	An enumerated value that indicates an administrator's default or startup configuration for the Enabled State of an element.	2 (Enabled)

Table 89. IBM_JobQueue (continued)

Property name	Property Description	Value or Value Location
string QueueStatusInfo	Provides a textual explanation for the status of the queue.	
uint32 JobsOnQueue	Current® number of jobs on queue.	

IBM_Subsystem

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

Table 90. IBM_Subsystem

Property name	Property Description	Value or Value Location
string CreationClassName (key)	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	IBM_Subsystem
string Name (key)	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	Subsystem library name/ Subsystem name
string Caption	A short textual description of the object.	Subsystem
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	Subsystem library name/ Subsystem name
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	
uint16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element.	5 (No Change)
uint16 EnabledDefault	An enumerated value that indicates an administrator's default or startup configuration for the Enabled State of an element.	2 (Enabled)
uint32 ActiveJobs	The number of jobs currently active in the subsystem.	
string Status	Provides a textual explanation for the status of the subsystem.	
sint32 MaxActiveJobs	The maximum number of jobs that can run or use resources in the subsystem at one time. If the subsystem description specifies *NOMAX, indicating that there is no maximum, this number is -1.	

Hardware inventory and network management providers

Hardware inventory providers collect information about hardware resources, hardware status, and hardware descriptions on the system. Network management providers collect information for network management. The information might be useful for upgrades and problem analysis.

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

Table 91. Supported providers

Provider name	Implements CIM class	Provider type
QUME_AdminDomainProvider	IBM_AdminDomain	Instance

Table 91. Supported providers (continued)

Provider name	Implements CIM class	Provider type
QUME_AssociatedMemoryProvider	IBM_AssociatedMemory	Instance and association
QUME_BindsToLANEndpointProvider	IBM_BindsToLANEndpoint	Instance and association
QUME_BIOSElementProvider	IBMi_BIOSElement	Instance
QUME_BusControllerProvider	IBMi_BusController	Instance
QUME_CardOnCardProvider	IBM_CardOnCard	Instance and association
QUME_CardProvider	IBM_Card	Instance
QUME_CDROMDriveProvider	IBMi_CDROMDrive	Instance
QUME_CentralEnclosureProvider	IBMi_CentralEnclosure	Instance
QUME_ChassisProvider	IBMi_Chassis	Instance
QUME_ChipProvider	IBM_Chip	Instance
QUME_CommIOPProvider	IBMi_CommIOP	Instance
QUME_CommPortProvider	IBMi_CommPort	Instance
QUME_CommPortImplementsLANEndpointProvider	IBMi_CommPortImplementsLANEndpoint	Instance and association
QUME_ComputerSystemPackageProvider	IBM_ComputerSystemPackage	Instance and association
QUME_ControlledByProvider	IBM_ControlledBy	Instance and association
QUME_ControlPanelProvider	IBMi_ControlPanel	Instance
QUME_CryptControllerProvider	IBMi_CryptController	Instance
QUME_CryptDeviceProvider	IBMi_CryptDevice	Instance
QUME_CryptIOPProvider	IBMi_CryptIOP	Instance
QUME_DeviceSAPImplementationProvider	IBM_DeviceSAPImplementation	Instance and association
QUME_DiskDriveProvider	IBMi_DiskDrive	Instance
QUME_DisplayProvider	IBMi_Display	Instance
QUME_DisplayControllerProvider	IBMi_DisplayController	Instance
QUME_DNSGeneralSettingDataProvider	IBM_DNSGeneralSettingData	Instance
QUME_DNSSettingDataProvider	IBM_DNSSettingData	Instance
QUME_DVDDriveProvider	IBMi_DVDDrive	Instance
QUME_ElementFRUProvider	IBM_ElementFRU	Instance and association
QUME_ElementSettingDataProvider	IBM_ElementSettingData	Instance and association

Table 91. Supported providers (continued)

Provider name	Implements CIM class	Provider type
QUME_EthernetPortProvider	IBM_EthernetPort	Instance
QUME_FileServerProvider	IBMi_FileServer	Instance
QUME_HostedAccessPointProvider	IBM_HostedAccessPoint	Instance and association
QUME_InstalledOSProvider	IBM_InstalledOS	Instance and association
QUME_IPProtocolEndpointProvider	IBM_IPProtocolEndpoint	Instance
QUME_LANEndpointProvider	IBM_LANEndpoint	Instance
QUME_LinkAggregatorPartnerConnectionProvider	IBMi_LinkAggregatorPartnerConnection	Instance and association
QUME_MemoryProvider	IBMi_Memory	Instance
QUME_PackagedComponentProvider	IBM_PackagedComponent	Instance and association
QUME_PackageInChassisProvider	IBM_PackageInChassis	Instance and association
QUME_PCIBridgeProvider	IBMi_PCIBridge	Instance
QUME_PCIControllerProvider	IBMi_PCIController	Instance
QUME_PCIDeviceProvider	IBM_PCIDevice	Instance
QUME_PhysicalMediaProvider	IBM_PhysicalMedia	Instance
QUME_PhysicalMemoryProvider	IBM_PhysicalMemory	Instance
QUME_PortControllerProvider	IBMi_PortController	Instance
QUME_PortImplementsEndpointProvider	IBM_PortImplementsEndpoint	Instance and association
QUME_PrinterProvider	IBMi_Printer	Instance
QUME_ProcessorProvider	IBMi_Processor	Instance
QUME_ProcessorCapacityProvider	IBMi_ProcessorCapacity	Instance
QUME_ProductPhysicalComponentProvider	IBM_ProductPhysicalComponent	Instance and association
QUME_ProductProvider	IBM_Product	Instance
QUME_RealizesProvider	IBM_Realizes	Instance and association
QUME_ReplacementFRUProvider	IBM_ReplacementFRU	Instance
QUME_SANBusProvider	IBMi_SANBus	Instance
QUME_ServiceProcessorProvider	IBMi_ServiceProcessor	Instance
QUME_SESDeviceProvider	IBMi_SESDevice	Instance
QUME_SNMPCommunityStringProvider	IBM_SNMPCommunityString	Instance
QUME_SNMPTrapTargetProvider	IBM_SNMPTrapTarget	Instance

Table 91. Supported providers (continued)

Provider name	Implements CIM class	Provider type
QUME_SOCIOPProvider	IBMi_SOCIOP	Instance
QUME_SPDBusProvider	IBMi_SPDBus	Instance
QUME_StaticIPAssignmentSettingDataProvider	IBM_StaticIPAssignmentSettingData	Instance
QUME_StorageControllerProvider	IBMi_StorageController	Instance
QUME_StorageIOPProvider	IBMi_StorageIOP	Instance
QUME_StoragePoolProvider	IBM_StoragePool	Instance
QUME_SystemDeviceProvider	IBM_SystemDevice	Instance and association
QUME_SystemPackagingProvider	IBM_SystemPackaging	Instance and association
QUME_TapeDriveProvider	IBMOS400_TapeDrive	Instance
QUME_TapeLibraryProvider	IBMi_TapeLibrary	Instance
QUME_TCPProtocolEndpointProvider	IBM_TCPProtocolEndpoint	Instance
QUME_TimeZoneSettingDataProvider	IBM_TimeZoneSettingData	Instance
QUME_TokenRingPortProvider	IBM_TokenRingPort	Instance
QUME_WirelessLANEndpointProvider	IBM_WirelessLANEndpoint	Instance
QUME_WirelessPortProvider	IBM_WirelessPort	Instance

IBM_AdminDomain

This provider represents a kind of network device net server. It includes some common properties a server has. This provider returns one instance of current IBM i as a net server.

Table 92. IBM_AdminDomain

Property name	Property description	Value or value location
string Name (key)	The name of this net server.	
string CreationClassName (key)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBM_AdminDomain
string Caption	The label by which the object is known.	
string Description	The label for some explanation for the object.	
string ElementName	A user-friendly name of the object.	
string NameFormat	A string that describes how the Name of the AdminDomain is generated.	AS
string DomainName	The domain name of this net server.	

Table 92. IBM_AdminDomain (continued)

Property name	Property description	Value or value location
string ServerDescription	The description of this net server.	

IBM_AssociatedMemory

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

Table 93. 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 94. 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.	

IBMi_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 95. IBMi_BIOSElement

Property name	Property description	Value or value location
string ActiveFixPackID	the level of server firmware on your system. The active copy is the level of server firmware that is currently running	

Table 95. IBMi_BIOSElement (continued)

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 FirmwareIDRelease	what IBM i product matches the level of the server firmware on your system. Managing the server firmware level is performed by applying or removing PTFs for this product	
string FirmwareUpdatePolicy	if the server firmware is currently being managed by the operating system or an HMC.	
string Manufacturer	CIM_BIOSElement.	IBM
string Name(key)(256)	The name that identifies this software element.	
string PermFixPackID	the level of server firmware on your system. The permanent copy is the backup level of server firmware	
string ServerFirmwareStatus	The status of the server firmware that is currently active	
string ServerIPLSource	the side (Temporary or Permanent) on which the last server IPL was performed for the server firmware code	
string ServicePartition	if current partition is a Service Partition	
string SoftwareElementID(256)	An identifier for the software element. The identifier is used with other keys to create a unique representation of the element.	
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.	OS400
string TempFixPackID	the level of server firmware on your system. The temporary copy is the installed level of server firmware	
string Version	Software version. It should be in the form <Major>.<Minor>.<Revision> or <Major>.<Minor><Letter><Revision>.	

IBMi_BusController

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

Table 96. IBMi_BusController

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 96. IBMi_BusController (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 96. IBMi_BusController (continued)		
Property name	**Property description**	**Value or value location**
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Bus Controller *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_BusController

IBM_CardOnCard

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

Table 97. 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 98. 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.	

Table 98. IBM_Card (continued)

Property name	Property description	Value or value location
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.	
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

IBMi_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 99. IBMi_CDROMDrive

Property name	**Property description**	**Value or value location**
boolean IsOEM	Original Equipment Manufacturer	
boolean MediaIsLocked	A property that indicates whether the media are locked and cannot be ejected.	TRUE
boolean ReportedIPL	Whether the resource reported this initial program load (IPL)	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card	
sint32 CardNumber	A numerical representation of the location of the card on the bus	
sint32 SessionNumber	The shared session number of the resource	
string CardID	The physical location where the device or feature is plugged into the bus	
string Caption (64)	A short textual description of the object.	CDROM *ElementName*
string ConsoleUsage	A value that indicates whether this resource is the primary console, the secondary console, or not used as a console of the system	
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_CDROMDrive
string CustomerCardID	Customer Card Identification Number	
string Description	A textual description of the object.	
string DeviceID(Key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string DevicePosition	The relative device position of the resource	

Table 99. IBMi_CDROMDrive (continued)

Property name	Property description	Value or value location
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which this resource is emulating	
string EmulatingType	The object type number that this resource is emulating	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full Logical Address	
string LocationCode	The physical location of the hardware resource in the system	
string LogicalAddress[]	Logical Address	
string LogicalCategories[]	Logical Categories	
string LogicalHierarchy[]	Logical Hierarchy	
string Model	Model Number	
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 PartNumber	A manufacturing identifier that represents similar types of hardware	
string PhysicalName	Assigned Physical Name	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier	
string ResourceKind[]	The resource kind field consists of 24 bytes of hexadecimal numbers	
string SerialNumber	The manufacturing sequence number of designation for the resource	
string SpecialIndicators[]	Special Indicators	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName(Key) (256)	The name of the scoping system.	
uint16 Availability	The primary availability and status of the device.	

Table 99. IBMi_CDROMDrive (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.	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 defined for the media access device.	2 (Unknown)

IBMi_CentralEnclosure

This provider returns an instance of the main card enclosure device that is available on the system when an enumerated list of instances is asked for, or the provider looks up the resource based on the logical resource name provided as the key under the DeviceID property.

Table 100. IBMi_CentralEnclosure

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	

Table 100. IBMi_CentralEnclosure (continued)

Property name	**Property description**	**Value or value location**
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	

Table 100. IBMi_CentralEnclosure (continued)

Property name	Property description	Value or value location
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Main Card Enclosure *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CentralEnclosure
string SystemFeatureCode	Feature code.	

IBM_Chassis

This provider is the same as IBMPSG_Chassis.

Related reference

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.

IBMi_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 101. ProviderTitle

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)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_Chassis
string Description	A textual description of the object.	Chassis information for *ElementName*

Table 101. ProviderTitle (continued)

Property name	**Property description**	**Value or value location**
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 SerialNumber (256)	A manufacturer-allocated number that is used to identify 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 OperationalStatus[]	The current statuses 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)
string ControllingID	System controlling ID.	
string FrameID	The identifier of a frame resource.	
string PowerDomainID	Power domain ID.	
string RackUniqueID	Rack unique ID.	
string TypeNumber	Machine type	
string LocationCode	The physical location of the hardware resource in the system	

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 102. 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.	
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.	

IBMi_CommIOP

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

Table 103. IBMi_CommIOP

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 103. IBMi_CommIOP (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 103. IBMi_CommIOP (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Communication I/O Processor *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CommIOP

IBMi_CommPort

This provider returns instances of all communication ports (excluding Ethernet ports, Token ring ports, and wireless ports) available on the system when an enumerated list of instances is asked for, or the provider looks up the resource based on the logical resource name provided as the key under the DeviceID property.

Table 104. IBMi_CommPort

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	

Table 104. IBMi_CommPort (continued)

Property name	Property description	Value or value location
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	

Table 104. IBMi_CommPort (continued)

Property name	Property description	Value or value location
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Communication Port *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CommPort
string NetworkAddresses [ ] (64)	An array of strings that indicates the network addresses for the port.	
uint16 PortNumber	NetworkPorts are often numbered relative to either a logical module or a network element.	
string PermanentAddress (64)	The network address that is hardcoded into a port.	

IBMi_CommPortImplementsLANEndpoint

This provider returns the association about the configuration of mapping relation between line description and port on the system.

Table 105. IBMi_CommPortImplementsLANEndpoint

Property name	Property value and data source	Instance mapping rule
IBMi_CommPort REF Antecedent	Returns a reference to the IBMi_CommPort, representing Ethernet port configuration data.	If IBM_LANEndpoint.Resourcename is *AGG: IBMi_CommPort.DeviceID mapping to IBM_LANEndpoint.EthernetDeviceIDList, otherwise IBMi_CommPort.DeviceID mapping to IBM_LANEndpoint.Name. The mapping numerical ratio should be one IBM_LANEndpoint mapping to n IBMi_CommPort.
IBM_LANEndpoint REF Dependent	Returns a reference to the IBM_LANEndpoint, representing Line description configuration data.	

IBM_ComputerSystemPackage

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

Table 106. 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 107. 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		

IBMi_ControlPanel

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

Table 108. IBMi_ControlPanel

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 108. IBMi_ControlPanel (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 108. IBMi_ControlPanel (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Control Panel *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_ControlPanel

IBMi_CryptController

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

Table 109. IBMi_CryptController

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	

Table 109. IBMi_CryptController (continued)

Property name	Property description	Value or value location
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	

Table 109. IBMi_CryptController (continued)

Property name	**Property description**	**Value or value location**
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Cryptographic Controller *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CryptController

IBMi_CryptDevice

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

Table 110. IBMi_CryptDevice

Property name	**Property description**	**Value or value location**
string Description	A textual description of the object.	

Table 110. IBMi_CryptDevice (continued)

Property name	Property description	Value or value location
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	

Table 110. IBMi_CryptDevice (continued)

Property name	Property description	Value or value location
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	

Table 110. IBMi_CryptDevice (continued)

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Cryptographic Device *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CryptDevice

IBMi_CryptIOP

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

Table 111. IBMi_CryptIOP

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	

Table 111. IBMi_CryptIOP (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 name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	

Table 111. IBMi_CryptIOP (continued)

Property name	**Property description**	**Value or value location**
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Cryptographic I/O Processor *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_CryptIOP

IBM_DeviceSAPImplementation

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

Table 112. 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.	

IBMi_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 113. IBMi_DiskDrive

Property name	**Property description**	**Value or value location**
boolean IsOEM	Original equipment manufacturer.	
boolean MediaIsLocked	A property that indicates whether the media are locked in the device and cannot be ejected.	TRUE
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	Disk Drive *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CreationClassName (Key) (256)	The name of the class or the subclass used in the creation of an instance.	IBMi_DiskDrive
string CustomerCardID	Customer card identification number.	
string Description	A textual description of the object.	
string DeviceID (Key)(64)	An address that uniquely names the logical device.	*Name*
string DevicePosition	The relative device position of the resource.	

Table 113. IBMi_DiskDrive (continued)

Property name	Property description	Value or value location
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
string Model	Model number.	
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 PartNumber	A manufacturing identifier that represents similar types of hardware.	
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key)(256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key)(256)	The name of the scoping system.	

Table 113. IBMi_DiskDrive (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 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).	

IBMi_Display

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

Table 114. IBMi_Display

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*

Table 114. IBMi_Display (continued)

Property name	Property description	Value or value location
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	

Table 114. IBMi_Display (continued)

Property name	**Property description**	**Value or value location**
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Work Station *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	

Table 114. IBMi_Display (continued)

Property name	Property description	Value or value location
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_Display

IBMi_DisplayController

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

Table 115. IBMi_DisplayController

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	

Table 115. IBMi_DisplayController (continued)

Property name	Property description	Value or value location
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	

Table 115. IBMi_DisplayController (continued)

Property name	Property description	Value or value location
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Display Controller *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_DisplayController

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 116. 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

Table 116. IBM_DNSGeneralSettingData (continued)

Property name	Property description	Value or value location
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 117. 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

IBMi_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 118. IBMi_DVDDrive

Property name	Property description	Value or value location
boolean IsOEM	Original equipment manufacturer.	
boolean MediaIsLocked	A property that indicates whether the media are locked in the device and cannot be ejected.	TRUE

Table 118. IBMi_DVDDrive (continued)

Property name	**Property description**	**Value or value location**
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	DVD Drive *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_DVDDrive
string CustomerCardID	Customer card identification number.	
string Description	A textual description of the object.	
string DeviceID(Key) (64)	An address that names the logical device.	*Name*
string DevicePosition	The relative device position of the resource.	
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	

Table 118. IBMi_DVDDrive (continued)

Property name	Property description	Value or value location
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
string Model	Model number.	
string Name (1024)	The label by which the object is known.	
string NetworkAddress	The network address of the LAN adapter resource.	
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 PartNumber	A manufacturing identifier that represents similar types of hardware.	
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName(Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName(Key) (256)	The name of the scoping system.	
uint16 Availability	The primary availability and status of the device.	

Table 118. IBMi_DVDDrive (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.	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 119. 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 120. 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 121. 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*

Table 121. IBM_EthernetPort (continued)

Property name	Property description	Value or value location
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 name of the scoping system.	
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.	

Table 121. IBM_EthernetPort (continued)

Property name	Property description	Value or value location
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.	
string AggregateStandard	The Link Aggregation Standard.	
string AggregatePolicy	The aggregate policy.	
uint16 AggregatedPortNumber	The aggregated Ethernet port number. If this line description does not configure aggregated port, the value should be 0.	
string EthernetDeviceIDList[]	The array list of aggregated Ethernet port names. If aggregated port number is 0, this list should be empty.	

IBMi_FileServer

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

Table 122. IBMi_FileServer

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	7

Table 122. IBMi_FileServer (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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	

Table 122. IBMi_FileServer (continued)

Property name	Property description	Value or value location
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	File Server *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_FileServer

IBM_HostedAccessPoint

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

Table 123. 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 124. 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_IPProtocolEndpoint

An IBM_IPProtocolEndpoint is mapped to a network interface on an i5/OS iSeries system. This provider returns instances of all network interfaces on the system when an enumerated list of instances is asked for, or looks up a resource based on the IP address, IP version and line name provided as the key under the name property.

Table 125. IBM_IPProtocolEndpoint

Property name	Property description	Value or value location
string Caption(64)	A short textual description of the object.	IP protocol endpoint Name
string CreationClassName(Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBM_IPProtocolEndpoint
string Description	A textual description of the object.	The IBM i IP protocol endpoint named Name, and associated with the line description *LineName*
string ElementName	A user-friendly name of the object.	*Name*

Table 125. IBM_IPProtocolEndpoint (continued)

Property name	Property description	Value or value location
string Name(256)	A string that identifies this protocol endpoint with either a port or an interface on a device.	The TCP interface IP address. For example 1.2.3.4.
string NameFormat(256)	The name that ensures that the value of the name property is unique.	<Internet Address>
string SystemCreationClassName (Key)(256)	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string SystemName(Key)(256)	The name of the scoping system.	
string IPv4Address	The IPv4 address that this ProtocolEndpoint represents.	
string IPv6Address	The IPv6 address that this ProtocolEndpoint represents.	
uint16 IPVersionSupport(Key)	This property explicitly defines support for different versions of the IP protocol, for this Endpoint.	0 (Unknown), 1 (IPv4 Only), 2 (IPv6 Only), 3 (Both IPv4 and IPv6)
string LineName(Key)	Name of the communications line description that identifies the physical network associated with an interface. This field is blank-padded.	
uint16[] OperationalStatus	Indicates the current statuses of the element.	
uint16 EnabledState	It is an integer enumeration that indicates the enabled and disabled states of an element.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1 ("Other"). This property must be set to null when EnabledState is any value other than 1.	
uint16 RequestedState	It is an integer enumeration that indicates the last requested or desired state for the element.	
uint16 EnabledDefault	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	

Table 125. IBM_IPProtocolEndpoint (continued)

Property name	Property description	Value or value location
uint16 ProtocolType	An enumeration that provides information to categorize and classify different instances of this class.	
string OtherTypeDescription	A string that describes the type of ProtocolEndpoint when the Type property of this class (or any of its subclasses) is set to 1 (Other). This property should be set to null when the Type property is any value other than 1.	
string AutoStart	Whether the interface is started automatically	*YES or *NO
string Status	Current status of the logical interface.	
string[] StatusDescriptions	Strings describing the various OperationalStatus array values.	
string SubnetMask	The mask for the IPv4 address of this ProtocolEndpoint, if one is defined.	
uint8 PrefixLength	The prefix length for the IPv6 address of this Protocol Endpoint, if one is defined.	

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 126. 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.	

Table 126. IBM_LANEndpoint (continued)

Property name	Property description	Value or value location
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
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.	
string ResourceName	The binding port name.	
string AggregateStandard	The Link Aggregation Standard.	
string AggregatePolicy	The aggregate policy.	
uint16 AggregatedPortNumber	The aggregated Ethernet port number. If this line description does not configure aggregated port, this value should be 0.	

Table 126. IBM_LANEndpoint (continued)		
Property name	**Property description**	**Value or value location**
string EthernetDeviceIDList[]	The array list of aggregated Ethernet port names. If aggregated port number is 0, this list should be empty.	

IBMi_LinkAggregatorPartnerConnection

This provider returns the association between aggregated line description and Ethernet Port.

Table 127. IBMi_LinkAggregatorPartnerConnection		
Property name	**Property value and data source**	**Instance mapping rule**
IBM_EthernetPort REF Antecedent	Returns a reference to the IBM_EthernetPort, representing Line description configuration data.	IBMi_CommPort.DeviceID mapping to IBM_EthernetPort.EthernetDeviceIDList. The mapping numerical ratio should be one IBM_EthernetPort mapping to n IBMi_CommPort.
IBMi_CommPort REF Dependent	Returns a reference to the IBMi_CommPort, representing Ethernet port configuration data.	

IBMi_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 128. IBMi_Memory		
Property name	**Property description**	**Value or value location**
boolean IsOEM	Original equipment manufacturer.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean Volatile	A property that indicates whether this memory is volatile.	TRUE
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	Cache Memory *ElementName*

Table 128. IBMi_Memory (continued)

Property name	Property description	Value or value location
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_Memory
string CustomerCardID	Customer card identification number.	
string Description	A textual description of the object.	
string DeviceID (Key) (64)	An address that names the logical device.	*ElementName*
string DevicePosition	The relative device position of the resource.	
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
string Model	Model number.	
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
string PartNumber	A manufacturing identifier that represents similar types of hardware.	

Table 128. IBMi_Memory (continued)

Property name	Property description	Value or value location
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
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 129. 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 130. 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.	

IBMi_PCIBridge

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

Table 131. IBMi_PCIBridge

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	

Table 131. IBMi_PCIBridge (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	

Table 131. IBMi_PCIBridge (continued)

Property name	Property description	Value or value location
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber_sint32	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	PCI Bridge *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_PCIBridge

IBMi_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 132. IBMi_PCIController

Property name	Property description	Value or value location
boolean IsOEM	Original equipment manufacturer.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	PCI Controller *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_PCIController
string CustomerCardID	Customer card identification number.	
string Description	A textual description of the object.	
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string DevicePosition	The relative device position of the resource.	
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which the resource is emulating.	

Table 132. IBMi_PCIController (continued)

Property name	**Property description**	**Value or value location**
string EmulatingType	The object type number that the resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
string Model	Model number.	
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 PartNumber	A manufacturing identifier that represents similar types of hardware.	
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
uint16 Availability	The primary availability and status of the device.	

Table 132. IBMi_PCIController (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.	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 133. 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

Table 133. IBM_PCIDevice (continued)

Property name	**Property description**	**Value or value location**
string SystemName (Key) (256)	The name of the scoping system.	
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 134. 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*

Table 134. IBM_PhysicalMedia (continued)

Property name	Property description	Value or value location
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.

See the information about the IBMPSG_PhysicalMemory class for details.

Related reference

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.

IBMi_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 135. IBMi_PortController

Property name	Property description	Value or value location
boolean IsOEM	Original equipment manufacturer.	

Table 135. IBMi_PortController (continued)

Property name	**Property description**	**Value or value location**
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	Port Controller *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CustomerCardID	Customer card identification number.	
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_PortController
string Description	A textual description of the object.	
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string DevicePosition	The relative device position of the resource.	
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	

Table 135. IBMi_PortController (continued)

Property name	Property description	Value or value location
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
string Model	Model number.	
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 PartNumber	A manufacturing identifier that represents similar types of hardware.	
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string SpecialIndicators[]	Special indicators	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
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.	

Table 135. IBMi_PortController (continued)

Property name	Property description	Value or value location
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 136. 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.	

IBMi_Printer

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

Table 137. IBMi_Printer

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	7

Table 137. IBMi_Printer (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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	

Table 137. IBMi_Printer (continued)

Property name	**Property description**	**Value or value location**
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Printer *DeviceID*
string DeviceID(key) (64)	An address or other identifying information to uniquely name the logical device.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_Printer

IBMi_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 138. IBMi_Processor

Property name	Property description	Value or value location
boolean IsOEM	Original equipment manufacturer.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 SessionNumber	The shared session number of the resource.	
string Caption (64)	A short textual description of the object.	Processor *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string CreationClassName (key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMi_Processor
string CustomerCardID	Customer card identification number.	
string Description	A textual description of the object.	
string DeviceID (key) (64)	An address or other identifying information to uniquely name the logical device.	*Name*
string DevicePosition	The relative device position of the resource.	
string ElementName	A user-friendly name of the object.	
string EmulatingModel	A model number for which the resource is emulating.	

Table 138. IBMi_Processor (continued)

Property name	Property description	Value or value location
string EmulatingType	The object type number that the resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full logical address.	
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.
string LocationCode	The physical location of the hardware resource in the system.	
string LogicalAddress[]	Logical address.	
string LogicalCategories[]	Logical categories.	
string LogicalHierarchy[]	Logical hierarchy.	
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 PartNumber	A manufacturing identifier that represents similar types of hardware.	
string PhysicalName	Assigned physical name.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Role	The role of the processor.	Central Processor
string SerialNumber	The manufacturing sequence number of designation for the resource.	

Table 138. IBMi_Processor (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string StatusDescriptions[]	Various OperationalStatus array values.	
string SystemCreationClassName (key) (256)	The scoping system's CreationClassName.	IBMPSG_ComputerSystem
string SystemName (key) (256)	The name of the scoping system.	
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)

Related reference
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.

IBMi_ProcessorCapacity

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

Table 139. IBMi_ProcessorCapacity

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 139. IBMi_ProcessorCapacity (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 139. IBMi_ProcessorCapacity (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Processor Capacity Card *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_ProcessorCapacity

IBM_ProductPhysicalComponent

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

Table 140. 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 141. 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.	

Table 141. IBM_Product (continued)		
Property name	**Property description**	**Value or value location**
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 142. 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 143. 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*

IBMi_SANBus

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

Table 144. IBMi_SANBus

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 144. IBMi_SANBus (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 144. IBMi_SANBus (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	SAN Bus *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_SANBus

IBMi_ServiceProcessor

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

Table 145. IBMi_ServiceProcessor

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	

Table 145. IBMi_ServiceProcessor (continued)

Property name	Property description	Value or value location
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	

Table 145. IBMi_ServiceProcessor (continued)

Property name	Property description	Value or value location
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Service Processor *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_ServiceProcessor

IBMi_SESDevice

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

Table 146. IBMi_SESDevice

Property name	Property description	Value or value location
string Description	A textual description of the object.	

Table 146. IBMi_SESDevice (continued)

Property name	Property description	Value or value location
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	

Table 146. IBMi_SESDevice (continued)

Property name	Property description	Value or value location
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	

Table 146. IBMi_SESDevice (continued)

Property name	Property description	Value or value location
string Caption (64)	A short textual description of the object.	Storage Enclosure Services *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_SESDevice

IBM_SNMPCommunityString

This provider contains the controlling information of accessing SNMP Service.

Table 147. 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
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 148. IBM_SNMPTrapTarget

Property name	Property description	Value or value location
String AccessInfo	The host address.	

Table 148. IBM_SNMPTrapTarget (continued)

Property name	Property description	Value or value location
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.	

IBMi_SOCIOP

This provider returns instances of all shared object clustering (SOC) I/O processors available on the system when an enumerated list of instances is asked for, or the provider looks up the resource based on the logical resource name provided as the key under the DeviceID property.

Table 149. IBMi_SOCIOP

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	

Table 149. IBMi_SOCIOP (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	

Table 149. IBMi_SOCIOP (continued)

Property name	Property description	Value or value location
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	SOC I/O Processor *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_SOCIOP

IBMi_SPDBus

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

Table 150. IBMi_SPDBus

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 150. IBMi_SPDBus (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 150. IBMi_SPDBus (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	SPD Bus *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_SPDBus

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 151. 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*
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

IBMi_StorageController

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

Table 152. IBMi_StorageController

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	

Table 152. IBMi_StorageController (continued)

Property name	Property description	Value or value location
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	

Table 152. IBMi_StorageController (continued)

Property name	Property description	Value or value location
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Storage Controller *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_StorageController

IBMi_StorageIOP

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

Table 153. IBMi_StorageIOP

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	

Table 153. IBMi_StorageIOP (continued)

Property name	**Property description**	**Value or value location**
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	

Table 153. IBMi_StorageIOP (continued)

Property name	**Property description**	**Value or value location**
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Storage I/O Processor *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_StorageIOP

IBM_StoragePool

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

Table 154. IBM_StoragePool

Property name	**Property description**	**Value or value location**
string Caption (64)	A short textual description of the object.	System Pool *DeviceID*

Table 154. IBM_StoragePool (continued)

Property name	Property description	Value or value location
string Description	A textual description of the object.	System Pool information for *DeviceID*
string ElementName	A user-friendly name of the object.	
string Name (1024)	The label by which the object is known.	*DeviceID*
string InstanceID(key)	Within the scope of the instantiating Namespace, this property opaquely and uniquely identifies an instance of the class.	
string PoolID	A unique name in the context of the system that identifies the pool.	
uint64 TotalManagedSpace	The total amount of capacity usable for the allocation of storage volumes, logical disks, or child storage pools.	

IBM_SystemDevice

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

Table 155. 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 156. 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.	

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 157. IBMOS400_TapeDrive

Property name	Property description	Value or value location
boolean IsOEM	Original Equipment Manufacturer	
boolean ReportedIPL	Whether the resource reported this initial program load (IPL).	
real32 MediaSize	The size of media in inches.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged	
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card	
sint32 CardNumber	A numerical representation of the location of the card on the bus	
sint32 SessionNumber	The shared session number of the resource	
string Caption (64)	A short textual description of the object.	Tape *ElementName*
string CardID	The physical location where the device or feature is plugged into the bus	
string ConsoleUsage	A value that indicates whether this resource is the primary console, the secondary console, or not used as a console of the system	
string CreationClassName (Key) (256)	The name of the class or the subclass that is used in the creation of an instance.	IBMOS400_TapeDrive
string CustomerCardID	Customer Card Identification Number	
string Description	A textual description of the object.	
string DeviceID (Key) (64)	An address that names the logical device.	*Name*
string DevicePosition	The relative device position of the resource	
string ElementName	A user-friendly name of the object.	*Name*
string EmulatingModel	A model number for which this resource is emulating	

Table 157. IBMOS400_TapeDrive (continued)

Property name	Property description	Value or value location
string EmulatingType	The object type number that this resource is emulating.	
string FrameID	The identifier of a frame resource	
string FullLogicalAddress[]	Full Logical Address	
string LocationCode	The physical location of the hardware resource in the system	
string LogicalAddress[]	Logical Address	
string LogicalCategories[]	Logical Categories	
string LogicalHierarchy[]	Logical Hierarchy	
string Model	Model Number	
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 PartNumber	A manufacturing identifier that represents similar types of hardware	
string PhysicalName	Assigned Physical Name	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier	
string ReadDensities[]	Tape Read Densities	
string ResourceKind[]	The resource kind field consists of 24 bytes of hexadecimal numbers	
string SerialNumber	The manufacturing sequence number of designation for the resource	
string SpecialIndicators[]	Special Indicators	
string StatusDescriptions[]	The various OperationalStatus array values.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string TapeCapabilities[]	Tape Capabilities	
string WriteDensities[]	Tape Write Densities	
uint16 Availability	The primary availability and status of the device.	

Table 157. IBMOS400_TapeDrive (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.	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)

IBMi_TapeLibrary

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

Table 158. IBMi_TapeLibrary

Property name	Property description	Value or value location
string Description	A textual description of the object.	
string ElementName	A user-friendly name of the object.	*DeviceID*
uint16 HealthState	The current health of the element.	
string Name (1024)	The label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	The current statuses of the element.	
string StatusDescriptions[]	The various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	7

Table 158. IBMi_TapeLibrary (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.	
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
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The name of the scoping system.	
string CustomerCardID	Customer card identification number.	
string CardID	The physical location where the device or feature is plugged into the bus.	
string Model	Model number.	
string PartNumber	A manufacturing identifier that represents similar types of hardware.	
string SerialNumber	The manufacturing sequence number of designation for the resource.	
string LocationCode	The physical location of the hardware resource in the system.	
string PhysicalName	Assigned physical name.	
string EmulatingModel	A model number for which the resource is emulating.	
string EmulatingType	The object type number that the resource is emulating.	
string LogicalHierarchy[]	Logical hierarchy.	
string LogicalCategories[]	Logical categories.	
string LogicalAddress[]	Logical address.	
string DevicePosition	The relative device position of the resource.	
sint32 BoardNumber	A numerical representation of a section of the bus into which the card is plugged.	

Table 158. IBMi_TapeLibrary (continued)

Property name	Property description	Value or value location
sint32 BusNumber	A numerical representation of the path connection of the system processor to the card.	
sint32 CardNumber	A numerical representation of the location of the card on the bus.	
sint32 PortNumber	The port number of the resource.	
sint32 SessionNumber	The shared session number of the resource.	
string RCTTLevel	The Reference Code Translation Table (RCTT) identifier.	
boolean ReportedIPL	A property that indicates whether the resource reported the initial program load.	
boolean IsOEM	Original equipment manufacturer.	
string ConsoleUsage	A property that indicates whether the resource is the primary console, the secondary console, or not used as a console of the system.	
string FullLogicalAddress[]	Full logical address.	
string SpecialIndicators	Special indicators, such as load source disk units, work station console IOPs, and non-reporting resources.	
string ResourceKind[]	The kind of the resource. The field consists of 24 bytes of hexadecimal numbers.	
string Caption (64)	A short textual description of the object.	Tape Library *DeviceID*
string DeviceID(key)	A user-friendly name of the object.	
string CreationClassName (Key) (256)	A property that indicates the name of the class or the subclass used in the creation of an instance.	IBMi_TapeLibrary

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 159. 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 160. 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.	

Table 160. IBM_TimeZoneSettingData (continued)

Property name	Property description	Value or value location
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 161. 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.	

Table 161. IBM_TokenRingPort (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 name of the scoping system.	
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.	
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 162. 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.	
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 163. 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 name of the scoping system.	
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

Table 163. IBM_WirelessPort (continued)

Property name	Property description	Value or value location
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

Supported CIM SMI-S HBA and HDR providers

In the IBM Universal Manageability Enablement for i 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.

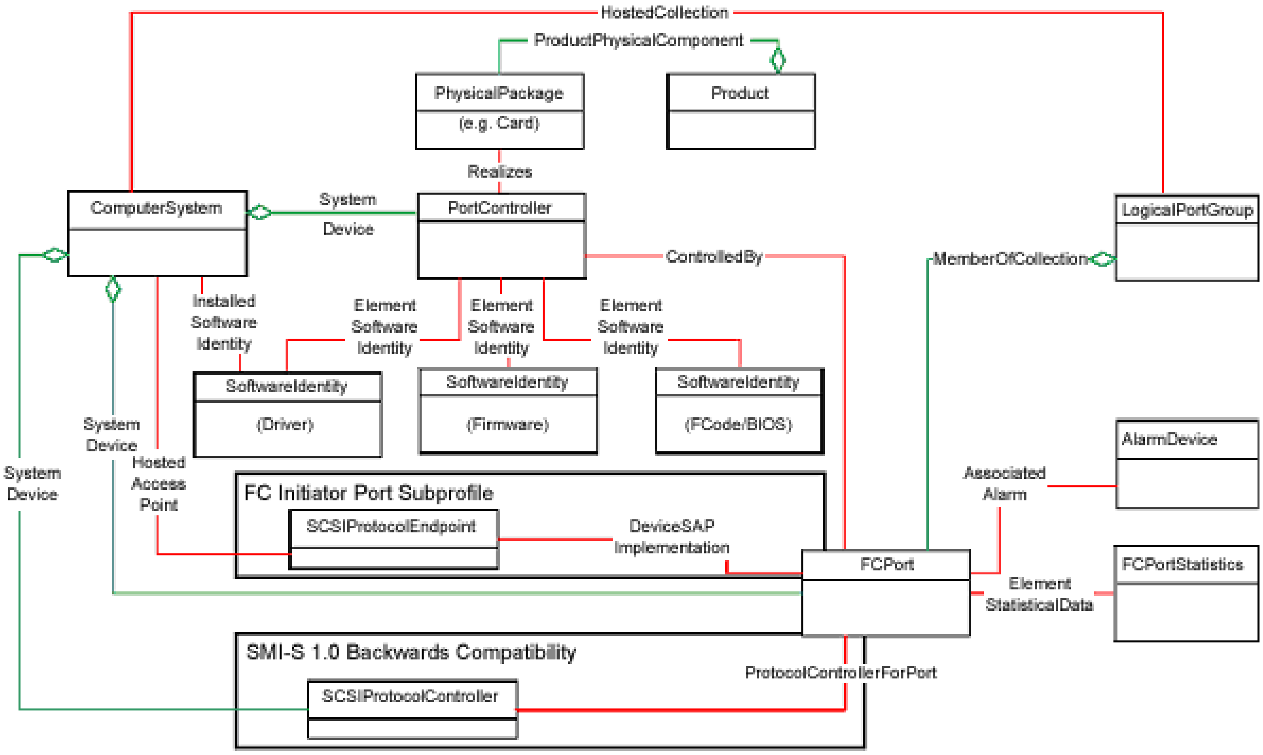


Figure 1. HBA profile

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.

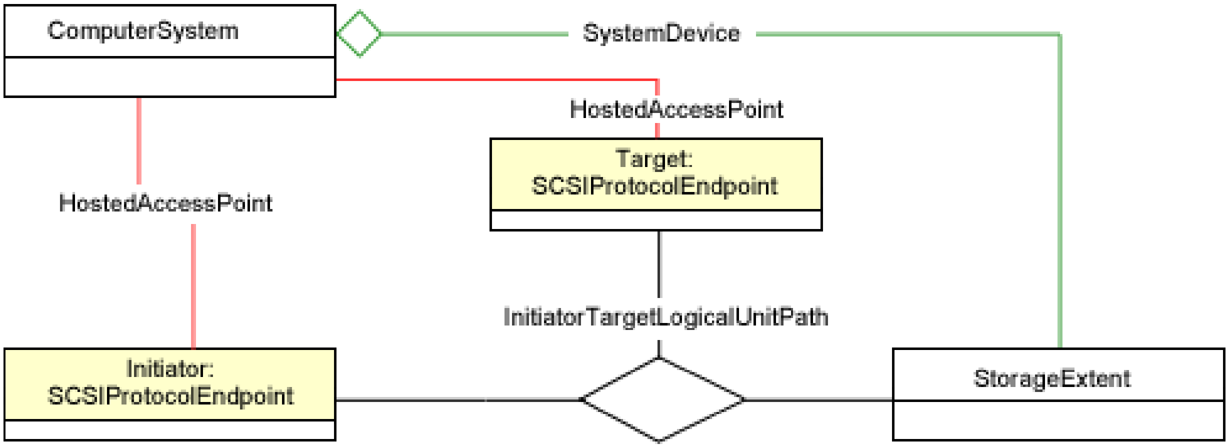
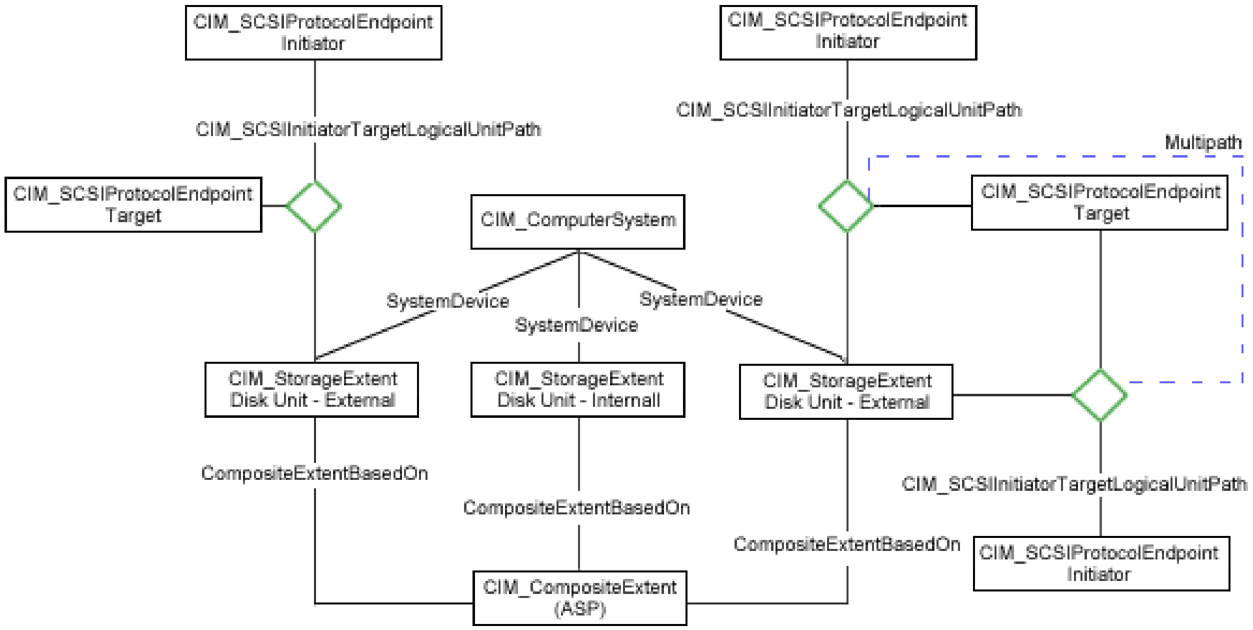


Figure 2. HDR profile

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

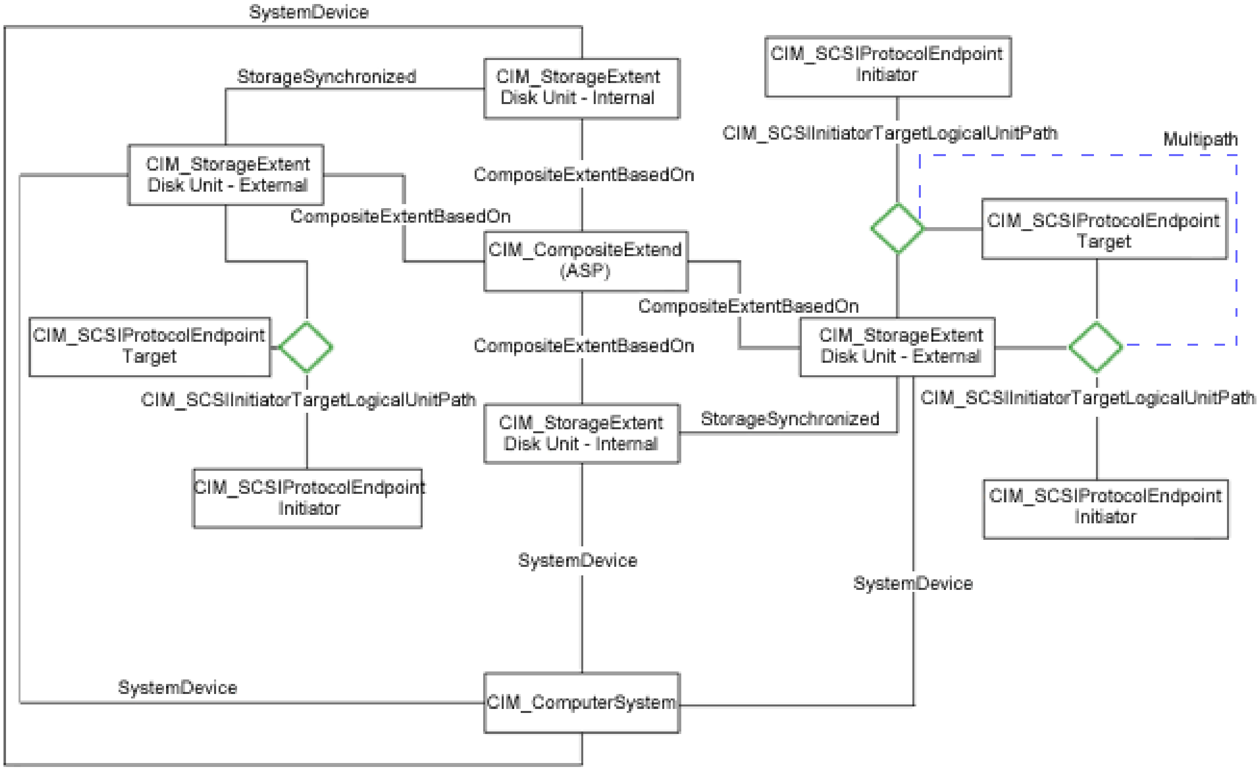
On IBM i, the HDR profile is extended to cover the requirement of modeling the virtual disk resource. The CompositeExtent class models the auxiliary storage pool (ASP), which may be consist of internal, external or virtual disk units. The StorageExtent class models the internal, external or virtual disk unit. The SCSIProtocolEndpoint class models the SCSI logical port, either initiator or target. As to the virtual storage, SCSIProtocolEndpoint represents the logical ports on virtual SCSI adapter.

HDR profiles are extended to model IBM i ASPs , mirroring, and multiple paths. Figure 3 and figure 4 show the modeling of IBM i 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 - Internal
CIM_StorageExtent Disk Unit - External
CompositeExtentBasedOn
CompositeExtentBasedOn
CIM_SCSIInitiatorTargetLogicalUnitPath
CompositeExtentBasedOn
CIM_SCSIProtocolEndpoint Initiator
CIM_CompositeExtent (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

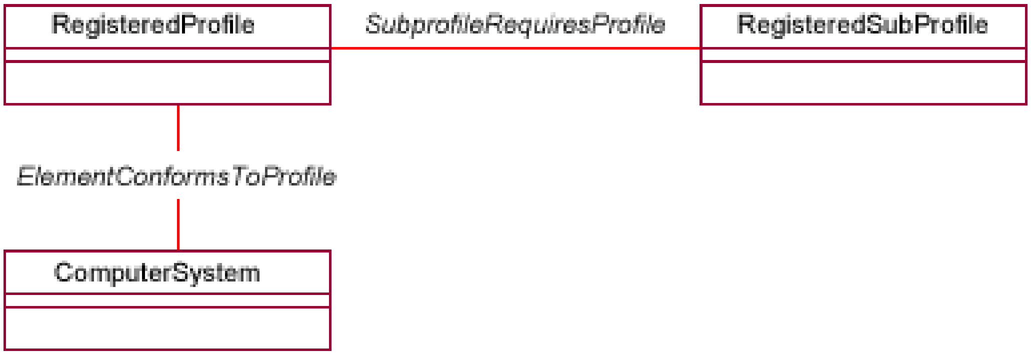


Figure 5. Registered profile

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

These IBM-supplied providers discover HBA card and storage resources on the host.

IBM_Card

This provider returns the HBA physical card instance on the system.

Table 164. 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.	

Table 164. 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 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

This provider is the same as IBMPSG_ComputerSystem.

Refer to the IBM_ComputerSystem provider in the "Providers that are inherited from the operating system" on page 46 topic for more information.

Related reference

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

IBM_FCPort

This 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 165. 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.	

Table 165. IBM_FCPort (continued)

Property name	Property description	Value or value location
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

This provider returns the association between a device and a controller.

Table 166. 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 167. 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 168. 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 169. 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 170. 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

Table 170. IBM_FCPortStatistics (continued)

Property name	Property description	Value or value location
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 171. 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 172. 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 173. 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 174. 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 175. 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

Table 175. IBM_PortController (continued)

Property name	Header	Header
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 176. 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 177. 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 ElementName	A user-friendly name for the object	For virtual SCSI: Adapter's resource name. For physical SCSI: Not implement
string Name (256)	The label by which the object is known.	
string OtherConnectionType	The connection type, if ConnectionType is 1 (Others).	Blank ("") or "Virtual SCSI"
string OtherTypeDescription (64)	The type of ProtocolEndpoint when the Type property of this class (or any of its subclasses) is set to 1.	Blank ("") or "Virtual Small Computer System Interface"
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 or 1 (Others)
uint16 ProtocolIFType	An enumeration that is synchronized with the IANA ifType MIB.	56 Fibre Channel or 1 (Other)
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 178. 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.

Table 178. IBM_SoftwareIdentity driver (continued)

Property name	Property description	Value or value location
string InstanceID (key)	Within the scope of the instantiating namespace, InstanceID identifies an instance of this class.	For IBM i version 6.1.0 or lower: SLIC_5761999. For IBM i version higher than 6.1.0: SLIC_5770999
string Manufacturer	The manufacturer of this software.	IBM
string VersionString	A string that represents the complete software version information.	SLIC version
uint16 []Classifications	An array of enumerated integers that classifies this software.	2 Driver

Table 179. 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 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.	
uint16 []Classifications	An array of enumerated integers that classifies this software.	10 (Firmware)

SMI-S HDR CIM classes

These SMI-S HDR 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

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

Table 180. 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 181. 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 (For ASP with virtual disk, set to null)
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

Table 181. IBM_CompositeExtent (continued)

Property name	Property description	Value or value location
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
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.	Page size
uint64 ConsumableBlocks	The maximum number of blocks that are available for consumption when layering StorageExtents using the BasedOn association.	Available 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 defines how data is striped across StorageExtents.

Table 182. 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 183. 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.	

Table 184. Instance Properties

Property name	Property description	Value or value location
unit32 AdministrativeWeight	A value assigned by an administrator specifying a preference to assign to a path. The drivers will actively use all available paths with the highest weight. This allows an administrator to assign a subset of available paths for load balanced access and reserve the others as backup paths. For symmetric access devices, all paths are considered 'available'. For asymmetric access devices, all paths in active target port groups are considered available.	0 (There is no path priority in IBM i)
uint32 State	The state of this path. Values are defined as follows: Unknown - the path is unavailable, but the cause is not known.	

Table 184. Instance Properties (continued)

Property name	Property description	Value or value location
uint16 AdministrativeOverride	AdministrativeOverride allows an administrator to select a single path, force all I/O to this path, and disables load balancing. The steady-state value is 'No override in effect'. When an administrator sets an override for a particular path, that path's AdministrativeOverride is set to 'Overridding' and all other paths to same logical unit are assigned a value of 'Overridden'. This property is changed using the OverridePath method in SCSIPathConfigurationService.	4 (No override in effect)

IBM_SCSIProtocolEndPoint

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

Table 185. 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.	Blank ("") or "Virtual Small Computer System Interface"
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) or 1 (Others)
string OtherConnectionType	The connection type, if ConnectionType is 1 (Others)	Blank ("") or "Virtual SCSI"
uint16 ProtocolIFType	An enumeration that is synchronized with the IANA ifType MIB.	56 (Fibre Channel) or 1 (Others)

Table 185. IBM_SCSIProtocolEndPoint (continued)

Property name	Property description	Value or value location
uint16 Role	For iSCSI, each SCSIProtocolEndpoint must act as either a target or an initiator endpoint.	3 Target
string ElementName	A user-friendly name for the object	For virtual SCSI: Adapter's resource name. For physical SCSI: Not implement

IBM_StorageExtent

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

Table 186. 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. (For virtual disk, set to null)
string Description	A textual description of the object.	This class represents instances of available disk units
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
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

Table 186. IBM_StorageExtent (continued)

Property name	Property description	Value or value location
string[] IdentifyingDescriptions	An array of freeform strings that provides explanations and details behind the entries in the OtherIdentifyingInfo array.	[0] - "Resource Name" [1] - "Logical Location Code" [2] - "Logical Unit Identifier", if LUN valid [3] - "World Wide Unique LUN ID"
string[] OtherIdentifyingInfo (256)	This property captures data besides the device ID information that can be used to identify a logical device.	[0] - Resource name [1] - Logical Location Code [2] - Logical Unit Identifier, if LUN valid [3] - World Wide Unique LUN ID, see **NOTE 1**
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

Table 186. IBM_StorageExtent (continued)

Property name	Property description	Value or value location
uint16 Access	This property describe whether the media is readable, writeable or both. Unknown and Write Once can also be defined.	
uint16 DataOrganization	The data organization type	2 (Fixed Block)

NOTE 1: The "World Wide Unique LUN ID" value is "UNSUPPORTED DEVICE TYPE" when the storage does not belong to external disk units under IOP-less fibre channel or virtual fibre channel (VFC) IOAs.

IBM_StorageSynchronized

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

Table 187. 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 IBM i.
IBM_StorageExtent REF SystemElement	The target of the replication.	

SMI-S Registered Profile CIM classes

These SMI-S Registered Profile CIM classes have been implemented as IBM-supplied providers to provide association information for discovered HBA card and storage resources on the host.

IBM_ElementConformsToProfile

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

Table 188. 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 189. 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*

Table 189. *IBM_RegisteredProfile (continued)*		
Property name	**Property description**	**Value or value location**
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 190. 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 191. 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.	

IBM i Base Metrics Providers

DMTF management profile, DSP 1053 Base Metrics Profile, is implemented on IBM i. The Base Metrics Profile is a component profile that defines the minimum object model needed to provide dynamic metrics associated to existing managed elements and related associations.

The following CIM classes have been implemented as IBM-supplied providers to provide Base Metrics Profile:

- IBM_HostedMetricService: a subclass of CIM_HostedService that associates between a computer system and metric service.
- IBM_MetricDefForME: a subclass of CIM_MetricDefForME that associates between a managed element (resource) and metric definition.
- IBM_MetricDefinition: a subclass of CIM_BaseMetricDefinition
- IBM_MetricForME: a subclass of CIM_MetricForME that associates between a managed element (resource) and metric value.
- IBM_MetricInstance: a subclass of CIM_MetricInstance that associates between metric definition and metric value.
- IBM_MetricService: a subclass of CIM_MetricService.
- IBM_MetricServiceAffectsElement: a subclass of CIM_ServiceAffectsElement that associates between a metric service and metric definition.
- IBM_MetricServiceCapabilities: a subclass of CIM_MetricServiceCapabilities.
- IBM_MetricServiceConformsToBaseMetricsProfile: a subclass of CIM_ElementConformsToProfile that associates between a metric registered profile and metric service.
- IBM_MetricServiceElementCapabilities: a subclass of CIM_ElementCapabilities that associates between a metric service and metric service capabilities.
- IBM_MetricValue: a subclass of CIM_BaseMetricValue
- IBM_RegisteredBaseMetricsProfile: a subclass of CIM_RegisteredProfile.

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

The following figure illustrates CIM standard schemas and IBM i extended metric classes.

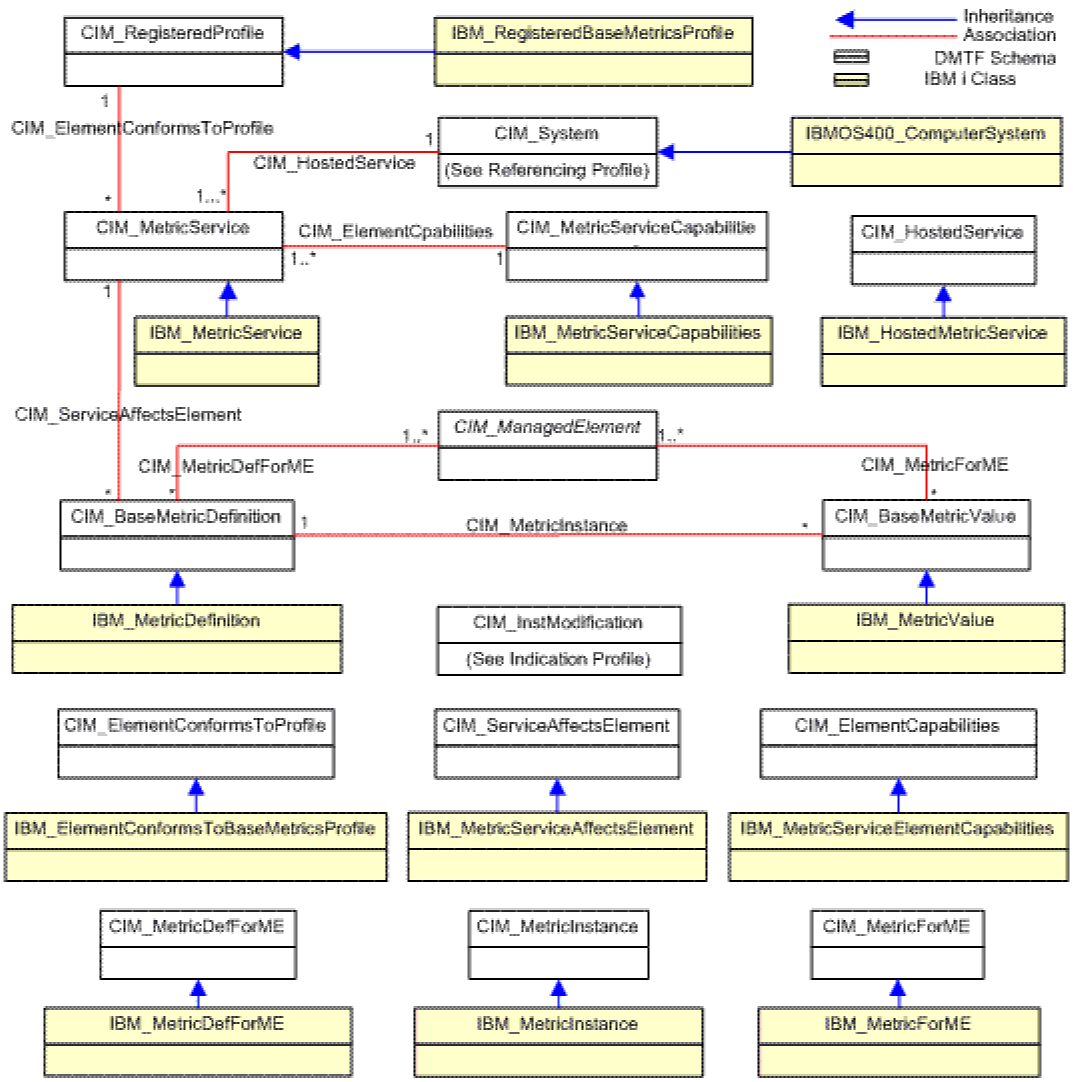


Figure 6. CIM metrics classes

The operating system supports the following CIM metrics.

Table 192. CIM base metrics providers

Provider Classr	**Implements CIM Class**	**Provider Type**
IBM_HostedMetricService	CIM_HostedService	Association
IBM_MetricDefForME	CIM_MetricDefForME	Association
IBM_MetricDefinition	CIM_BaseMetricDefinition	Instance
IBM_MetricForME	CIM_MetricForME	Association
IBM_MetricInstance	CIM_MetricInstance	Association
IBM_MetricService	CIM_MetricService	Instance
IBM_MetricServiceAffectsElement	CIM_ServiceAffectsElement	Association
IBM_MetricServiceCapabilities	CIM_MetricServiceCapabilities	Instance
IBM_MetricServiceConformsToBaseMetricsProfile	CIM_ElementConformsToProfile	Association
IBM_MetricServiceElementCapabilities	CIM_ElementCapabilities	Association
IBM_MetricValue	CIM_BaseMetricValue	Instance
IBM_MetricValueModification	CIM_InstModification	Indication

Table 192. CIM base metrics providers (continued)

Provider Classr	Implements CIM Class	Provider Type
IBM_RegisteredBaseMetricsProfile	CIM_RegisteredProfile	Instance

IBM_RegisteredBaseMetricsProfile

Description

IBM_RegisteredBaseMetricsProfile models the instances of Base Metrics Profile. Capabilities and management of profiles registered for Base Metrics profile

Hierarchy

CIM_RegisteredProfile, CIM_ManagedElement

Intrinsic Methods

getInstance, enumerateInstances, enumerateInstanceNames

Property Name	Class Defining Property	Property Description	Value or Value Location
string **RegisteredName**	CIM_RegisteredProfile	The name of this registered profile	"Base Metrics Profile"
string **RegisteredVersion**	CIM_RegisteredProfile	The version of this profile	"1.0.0"
uint16 **AdvertiseTypes**[ ]	CIM_RegisteredProfile	This property signifies the advertisement for the profile information	3 - "SLP"
uint16 **RegisteredOrganization**	CIM_RegisteredProfile	The organization that defines this profile.	2 - "DMTF"
string **InstanceID**	CIM_RegisteredProfile	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	"DSP1053 Base Metrics Profile, Version: 1.0.0a"

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.

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.

Three categories of metrics providers are implemented on IBM i. And CIM indication mechanism is introduced to notify applications when a specific metric event occurs. The following CIM class has been implemented as IBM-supplied provider to support the metrics value modification indications:

- IBM_MetricValueModification: a subclass of CIM_InstModification.

Related concepts

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.

Related tasks

Configuring CIMOM

Related information

User Function Registration APIs

IBM i File Metrics Providers

These providers provide capabilities to retrieve file statistics data on IBM i.

The file metrics providers provide the capabilities to retrieve file statistics data on IBM I realtimely. You can use these providers to get file size and last modification datetime.

Table 193. IBM i File Metrics

Resource/Base CIM class	Metric
CIM_LogicalFile **Note:** Since the CIM_LogicalFile is not implemented on IBM i, prohibit the client traverse those associations through CIM_MetricDefForME, CIM_MetricForME and CIM_MetricInstance associations on metrics model	**FileSize** File size in bytes, id: MDFL40 **LastModificationDateTime** Last modification datetime of the file, id: MDFL41

IBM_FileMetricServiceConformsToBaseMetricsProfile

Description: IBM_ElementConformsToBaseMetricsProfile association defines the RegisteredProfiles to which the referenced ManagedElement is conformant. This association may apply to any Managed Element. Typical usage will apply it to a higher level instance, such as a System, NameSpace, or Service. When applied to a higher level instance, all constituent parts MUST behave appropriately in support of the ManagedElement's conformance to the named RegisteredProfile.

Hierarchy: IBM_MetricServiceConformsToBaseMetricsProfile, CIM_ElementConformsToProfile

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Note: There are some limitations for cross-namespace association employed by Pegasus Server (See Open Pegasus PEP#139 -- Cross Namespace Associations). Operation can be navigated only in a unidirectional way. And the association was only registered to namespace root/PG_InterOp, so associators/associatornames/references/referencenames from IBM_MetricService are not supporteded.

Property name	Property value (& data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The ManagedElement that conforms to the RegisteredProfile	RegisteredProfile “Base Metrics Profile” associates to FileMetricService instance.
CIM_RegisteredProfile REF **ConformantStandard**	The RegisteredBaseMetricsProfile to which the MetricService conforms to.	

IBM_FileMetricServiceCapabilities

Description: Capabilities of a CIM_MetricService.

Hierarchy: IBM_MetricServiceCapabilities, CIM_MetricServiceCapabilities, CIM_EnabledLogicalElementCapabilities, CIM_Capabilities, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property name	Class Defining Property	Property Description	Value or Value Location
uint16 MetricsControlTypes[]	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_BaseMetricDefinition identified by the value at the same array index in the ControllableMetrics property.	0 - Unknown
uint16 **ManagedElementControlTypes[]**	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_ManagedElement identified by the value at the same array index in the ControllableManagedElements property.	0 - Unknown
boolean **ElementNameEditSupported**	CIM_EnabledLogicalElementCapabilities	Boolean indicating whether the ElementName can be modified.	FALSE
string **InstanceID**	CIM_Capabilities	Within the scope of the instantiating Namespace, this property opaquely and uniquely identifies an instance of this class.	"FileMetricServiceCapabilities"
String **ElementName**	CIM_Capabilities	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. **Note:** Name does not have to be unique within a namespace.	"Metric Service Capabilities"

IBM_FileMetricService

Description: The MetricService provides the ability to manage metrics.

Hierarchy: IBM_MetricService, CIM_MetricService, CIM_Service, CIM_EnabledLogicalElement, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property name	Class Defining Property	Property Description	Value or Value Location
string **SystemCreationClassName**	CIM_Service	The CreationClassName of the scoping System.	"IBMOS400_ComputerSystem"
string **SystemName**	CIM_Service	The Name of the scoping System.	system name
string **CreationClassName**	CIM_Service	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	"IBM_FileMetricService"
string **Name**	CIM_Service	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	"FileMetricService"
boolean **Started**	CIM_Service	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	TRUE

IBM_FileMetricServiceElementCapabilities

Description: IBM_MetricServiceElementCapabilities represents the association between ManagedElements and their Capabilities. The cardinality of the ManagedElement reference is Min(1), Max(1). This cardinality mandates the instantiation of the ElementCapabilities association for the referenced instance of Capabilities. ElementCapabilities describes the existence requirements and context for the referenced instance of ManagedElement. Specifically, the ManagedElement MUST exist and provides the context for the Capabilities.

Hierarchy: IBM_MetricServiceElementCapabilities,CIM_ElementCapabilities

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (& data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The managed element.	FileMetricService to FileMetricServiceCapabilities
CIM_Capabilities REF **Capabilities**	The Capabilities object associated with the element.	

IBM_HostedFileMetricService

Description: IBM_HostedMetricService is an association between a Service and the System on which the functionality is located.

Hierarchy: IBM_HostedMetricService, CIM_HostedService, CIM_HostedDependency, CIM_Dependency

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_Service REF **Dependent**	The Service hosted on the System.	Host System to FileMetricService.
CIM_System REF **Antecedent**	The hosting System.	

IBM_FileMetricServiceAffectsElement

Description: To represents an association between a Service and the ManagedElements that might be affected by its execution.

Hierarchy: IBM_MetricServiceAffectsElement,CIM_ServiceAffectsElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_ManagedElement REF **AffectedElement**	The Managed Element that is affected by the Service.	IBM_FileMetricDefinition to IBM_FileMetricService.
CIM_Service REF **AffectingElement**	The Service that is affecting the ManagedElement.	

Property name	Property value (and data source)	Instance mapping rule
uint16 **ElementEffects[]**	An enumeration that describes the effect on the ManagedElement. This array corresponds to the OtherElementEffectsDescriptions array, where the latter provides details that are related to the high-level effects enumerated by this property.	5:Manages

IBM_FileMetricDefinition

Description: An IBM_FileMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_FileMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. IBM_FileMetricDefinition instance represents the definition aspects of a metric on IBM i.

Hierarchy: IBM_MetricDefinition,CIM_BaseMetricDefinition, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property name	Class Defining Property	Property Description	Value or Value Location
string **Id**	CIM_BaseMetricDefinition	A string that uniquely identifies the metric definition. The use of OSF UUID/GUIDs is recommended.	Metric definition Id
string **Name**	CIM_BaseMetricDefinition	The name of the metric. This name does not have to be unique, but should be descriptive and may contain blanks.	Metric name
uint16 **DataType**	CIM_BaseMetricDefinition	The data type of the metric.	Metric data type
uint16 **Calculable**	CIM_BaseMetricDefinition	An enumerated value that describes the characteristics of the metric, for purposes of performing calculations	Metric calculable
string **Units**	CIM_BaseMetricDefinition	identifies the specific units of a value, like Bytes or Packets	Metric units
boolean **IsContinuous**	CIM_BaseMetricDefinition	IsContinuous indicates whether or not the metric value is continuous or scalar. Performance metrics are an example of a linear metric	Is Metric continuous
uint16 **ChangeType**	CIM_BaseMetricDefinition	ChangeType indicates how the metric value changes, in the form of typical combinations of finer grain attributes such as direction change, minimum and maximum values, and wrapping semanticst.	Metric change type
uint16 **TimeScope**	CIM_BaseMetricDefinition	TimeScope indicates the time scope to which the metric value applies.	Metric time scope
uint16 **GatheringType**	CIM_BaseMetricDefinition	GatheringType indicates how the metric values are gathered by the underlying instrumentation. This allows the client application to choose the right metric for the purpose.	Metric gathering type

Property name	Class Defining Property	Property Description	Value or Value Location
string **ElementName**	CIM_ManagedElement	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	Metric element name
string **Caption**	CIM_ManagedElement	The Caption property is a short textual description (one- line string) of the object.	Metric caption
string **Description**	CIM_ManagedElement	The Description property provides a textual description of the object.	Metric description

IBM_FileMetricValue

Description: IBM_FileMetricValue represents a metric value.

Hierarchy: IBM_MetricValue,CIM_BaseMetricValue, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property name	Class Defining Property	Property Description	Value or Value Location
string **InstanceID**	CIM_BaseMetricValue	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	This property of a metric value class must be generated and be unique for every instance (in time, duration, resource, and metric). **Note:** See NOTE 1
string **MetricDefinitionId**	CIM_BaseMetricValue	The key of the BaseMetricDefinition instance for this CIM_BaseMetricValue instance value	key defined for metric definition class
string **MeasuredElementName**	CIM_BaseMetricValue	A descriptive name for the element to which the metric value belongs (i.e., the measured element).	A short descriptive name for the managed element being measured as determined by the base class support. **Note:** See NOTE 2
datetime **TimeStamp**	CIM_BaseMetricValue	Identifies the time when the value of a metric instance is computed. Note that this is different from the time when the instance is created.	The date and time of the sample interval.
datetime **Duration**	CIM_BaseMetricValue	Property that represents the time duration over which this metric value is valid.	Interval seconds from our DB file except for point metrics which will be returned as zero
String **MetricValue**	CIM_BaseMetricValue	The value of the metric represented as a string. Its original data type is specified in CIM_BaseMetricDefinition.	Metric value
boolean **Volatile**	CIM_BaseMetricValue	If true, Volatile indicates that the value for the next point in time may use the same object and just change its properties (such as the value or timestamp). If false, the existing objects remain unchanged and a new object is created for the new point in time.	TRUE

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Resource identifier : • File – absolute file path

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- File - absolute file path

IBM_FileMetricInstance

Description: Weak association of metric value objects with their metric definition. This association ties an instance of CIM_BaseMetricValue to its CIM_BaseMetricDefinition; it is weak because the existence of a CIM_BaseMetricValue depends on the existence of its associated CIM_BaseMetricDefinition: An instance of CIM_BaseMetricValue must not exist without being associated to its corresponding CIM_BaseMetricDefinition.

Hierarchy: IBM_MetricInstance, CIM_MetricInstance

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A CIM_BaseMetricValue instance holding the Value.	IBM_FileMetricDefinition to IBM_FileMetricValue
CIM_BaseMetricDefinition REF **Antecedent**	The CIM_BaseMetricDefinition for this particular CIM_BaseMetricValue.	

IBM_FileMetricDefForME

Description: This association ties a CIM_BaseMetricDefinition to a CIM_ManagedElement to define metrics for the latter. The metrics definition is given context by the ManagedElement, which is why the definition is dependent on the element.

Hierarchy: IBM_MetricDefForME,CIM_MetricDefForMe

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_BaseMetricDefinition REF **Dependent**	A CIM_BaseMetricDefinition for a CIM_ManagedElement.	IBM_FileMetricDefinition to related managed element
CIM_ManagedElement REF **Antecedent**	The CIM_ManagedElement that can have metrics of this type associated with it.	

IBM_FileMetricForME

Description: This association links a ManagedElement to the metric values being maintained for it.

Hierarchy: IBM_MetricForME, CIM_MetricForME

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A metric value for the ManagedElement.	IBM_FileMetricValue to related managed element
CIM_ManagedElement REF **Antecedent**	ManagedElement to which the metric values belong.	

IBM i Spooled File Metrics Providers

These providers provide capabilities to retrieve spooled file statistics data on IBM i.

The file metrics providers provide the capabilities to retrieve spooled file statistics data on IBM i realtimely. You can use these providers to get spooledfile number on system.

Table 194. IBM i Spooled File Metrics

Resource/Base CIM Class	Metric
Individual Output Queues **Note:** Since the “Individual Output Queue” is not implemented on IBM i, prohibit the client traverse those associations through CIM_MetricDefForME and CIM_MetricForME associations on metrics model	**SpooledFiles** The number of spooled files on individual output queues, id: MDOQ40
ASP Group **Note:** Since the “ASP Group” is not implemented on IBM i, prohibit the client traverse those associations through CIM_MetricDefForME and CIM_MetricForME associations on metrics model	**SpooledFiles** The number of spooled files in ASP group, id: MDSP40

IBM_SpooledFileMetricServiceConformsToBaseMetricsProfile

Description: IBM_ElementConformsToBaseMetricsProfile association defines the RegisteredProfiles to which the referenced ManagedElement is conformant. This association may apply to any Managed Element. Typical usage will apply it to a higher level instance, such as a System, NameSpace, or Service. When applied to a higher level instance, all constituent parts MUST behave appropriately in support of the ManagedElement's conformance to the named RegisteredProfile.

Hierarchy: IBM_MetricServiceConformsToBaseMetricsProfile, CIM_ElementConformsToProfile

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Note: There are some limitations for cross-namespace association employed by Pegasus Server (See Open Pegasus PEP#139 -- Cross Namespace Associations). Operation can be navigated only in a unidirectional way. And the association was only registered to namespace root/PG_InterOp, so associators/associatornames/references/referencenames from IBM_MetricService are not supporteded.

Property Name	Property Value (& data source)	Instance Mapping Rule
CIM_ManagedElement REF **ManagedElement**	The ManagedElement that conforms to the RegisteredProfile.	RegisteredProfile “Base Metrics Profile” associates to SpooledFileMetricService instance.
CIM_RegisteredProfile REF **ConformantStandard**	The RegisteredBaseMetricsProfile to which the MetricService conforms to.	

IBM_SpooledFileMetricServiceCapabilities

Description: Capabilities of a CIM_MetricService.

Hierarchy: IBM_MetricServiceCapabilities, CIM_MetricServiceCapabilities, CIM_EnabledLogicalElementCapabilities, CIM_Capabilities, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
uint16 **MetricsControlTypes**[]	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_BaseMetricDefinition identified by the value at the same array index in the ControllableMetrics property.	0 - Unknown
uint16 **ManagedElementControlTypes**[]	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_ManagedElement identified by the value at the same array index in the ControllableManagedElements property.	0 - Unknown
boolean **ElementNameEditSupported**	CIM_EnabledLogicalElementCapabilities	Boolean indicating whether the ElementName can be modified.	FALSE
string **InstanceID**	CIM_Capabilities	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	"SpooledFileMetricServiceCapabilities"
String **ElementName**	CIM_Capabilities	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. **Note:** Name does not have to be unique within a namespace.	"Metric Service Capabilities"

IBM_SpooledFileMetricService

Description: The MetricService provides the ability to manage metrics.

Hierarchy: IBM_MetricService, CIM_MetricService, CIM_Service, CIM_EnabledLogicalElement, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
string **SystemCreationClassName**	CIM_Service	The CreationClassName of the scoping System.	"IBMOS400_ComputerSystem"
string **SystemName**	CIM_Service	The Name of the scoping System.	system name

Property Name	class Defining Property	Property Description	Value or Value Location
string **CreationClassName**	CIM_Service	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	"IBM_SpooledFileMetricService"
string **Name**	CIM_Service	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	"SpooledFileMetricService"
boolean **Started**	CIM_Service	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	TRUE

IBM_SpooledFileMetricServiceElementCapabilities

Description: IBM_MetricServiceElementCapabilities represents the association between ManagedElements and their Capabilities. The cardinality of the ManagedElement reference is Min(1), Max(1). This cardinality mandates the instantiation of the ElementCapabilities association for the referenced instance of Capabilities. ElementCapabilities describes the existence requirements and context for the referenced instance of ManagedElement. Specifically, the ManagedElement MUST exist and provides the context for the Capabilities.

Hierarchy: IBM_MetricServiceElementCapabilities,CIM_ElementCapabilities

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The managed element	SpooledFileMetricService to SpooledFileMetricServiceCapabilities
CIM_Capabilities REF **Capabilities**	The Capabilities object associated with the element.	

IBM_HostedSpooledFileMetricService

Description: IBM_HostedMetricService is an association between a Service and the System on which the functionality is located.

Hierarchy: IBM_HostedMetricService, CIM_HostedService, CIM_HostedDependency, CIM_Dependency

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property Value (and data source)	Instance mapping rule
CIM_Service REF **Dependent**	The Service hosted on the System.	Host System to SpooledFileMetricService.
CIM_System REF **Antecedent**	The hosting System	

IBM_SpooledFileMetricServiceAffectsElement

Description: To represents an association between a Service and the ManagedElements that might be affected by its execution.

Hierarchy: IBM_MetricServiceAffectsElement,CIM_ServiceAffectsElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF **AffectedElement**	The Managed Element that is affected by the Service	IBM_SpooledFileMetricDefinition to IBM_SpooledFileMetricService
CIM_Service REF **AffectingElement**	The Service that is affecting the ManagedElement.	
uint16 **ElementEffects**[]	An enumeration that describes the effect on the ManagedElement. This array corresponds to the OtherElementEffectsDescriptions array, where the latter provides details that are related to the high-level effects enumerated by this property.	5:Manages

IBM_SpooledFileMetricDefinition

Description: An IBM_SpooledFileMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_SpooledFileMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. IBM_SpooledFileMetricDefinition instance represents the definition aspects of a metric on IBM i.

Hierarchy: IBM_MetricDefinition,CIM_BaseMetricDefinition, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
string **Id**	CIM_BaseMetricDefinition	A string that uniquely identifies the metric definition. The use of OSF UUID/GUIDs is recommended.	Metric definition Id
string **Name**	CIM_BaseMetricDefinition	The name of the metric. This name does not have to be unique, but should be descriptive and may contain blanks.	Metric name

Property Name	**class Defining Property**	**Property Description**	**Value or Value Location**
uint16 **DataType**	CIM_BaseMetricDefinition	The data type of the metric.	Metric data type
uint16 **Calculable**	CIM_BaseMetricDefinition	An enumerated value that describes the characteristics of the metric, for purposes of performing calculations	Metric calculable
string **Units**	CIM_BaseMetricDefinition	identifies the specific units of a value, like Bytes or Packets	Metric units
boolean **IsContinuous**	CIM_BaseMetricDefinition	IsContinuous indicates whether or not the metric value is continuous or scalar. Performance metrics are an example of a linear metric	Is Metric continuous
uint16 **ChangeType**	CIM_BaseMetricDefinition	ChangeType indicates how the metric value changes, in the form of typical combinations of finer grain attributes such as direction change, minimum and maximum values, and wrapping semantics	Metric change type
uint16 **TimeScope**	CIM_BaseMetricDefinition	TimeScope indicates the time scope to which the metric value applies.	Metric time scope
uint16 **GatheringType**	CIM_BaseMetricDefinition	GatheringType indicates how the metric values are gathered by the underlying instrumentation. This allows the client application to choose the right metric for the purpose.	Metric gathering type
string **ElementName**	CIM_ManagedElement	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	Metric element name
string **Caption**	CIM_ManagedElement	The Caption property is a short textual description (one- line string) of the object.	Metric caption

Property Name	class Defining Property	Property Description	Value or Value Location
string **Description**	CIM_ManagedElement	The Description property provides a textual description of the object.	Metric description

IBM_SpooledFileMetricValue

Description: IBM_SpooledFileMetricValue represents a metric value.

Hierarchy: IBM_MetricValue,CIM_BaseMetricValue, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
string **InstanceID**	CIM_BaseMetricValue	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class	This property of a metric value class must be generated and be unique for every instance (in time, duration, resource, and metric). See **NOTE 1**
string **MetricDefinitionId**	CIM_BaseMetricValue	The key of the BaseMetricDefinition instance for this CIM_BaseMetricValue instance value	key defined for metric definition class
string **MeasuredElementName**	CIM_BaseMetricValue	A descriptive name for the element to which the metric value belongs (i.e., the measured element).	A short descriptive name for the managed element being measured as determined by the base class support. See **NOTE 2**
datetime **TimeStamp**	CIM_BaseMetricValue	Identifies the time when the value of a metric instance is computed. Note that this is different from the time when the instance is created.	The date and time of the sample interval.
datetime **Duration**	CIM_BaseMetricValue	Property that represents the time duration over which this metric value is valid.	Interval seconds from our DB file except for point metrics which will be returned as zero
String **MetricValue**	CIM_BaseMetricValue	The value of the metric represented as a string. Its original data type is specified in CIM_BaseMetricDefinition.	Metric value
boolean **Volatile**	CIM_BaseMetricValue	If true, Volatile indicates that the value for the next point in time may use the same object and just change its properties (such as the value or timestamp). If false, the existing objects remain unchanged and a new object is created for the new point in time.	TRUE

IBM_SpooledFileMetricInstance

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Dexcription
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Resource identifier: • Individual Output Queue: Queue Name • ASP Group: ASP Group Name

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- Individual Output Queue: Queue Name
- ASP Group: ASP Group Name

Description: Weak association of metric value objects with their metric definition. This association ties an instance of CIM_BaseMetricValue to its CIM_BaseMetricDefinition; it is weak because the existence of a CIM_BaseMetricValue depends on the existence of its associated CIM_BaseMetricDefinition: An instance of CIM_BaseMetricValue must not exist without being associated to its corresponding CIM_BaseMetricDefinition.

Hierarchy: IBM_MetricInstance, CIM_MetricInstance

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A CIM_BaseMetricValue instance holding the Value.	IBM_SpooledFileMetricDefinition to IBM_SpooledFileMetricValue
CIM_BaseMetricDefinition REF **Antecedent**	The CIM_BaseMetricDefinition for this particular CIM_BaseMetricValue.	

IBM_SpooledFileMetricDefForME

Description: This association ties a CIM_BaseMetricDefinition to a CIM_ManagedElement to define metrics for the latter. The metrics definition is given context by the ManagedElement, which is why the definition is dependent on the element.

Hierarchy: IBM_MetricDefForME,CIM_MetricDefForMe

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricDefinition REF **Dependent**	A CIM_BaseMetricDefinition for a CIM_ManagedElement.	IBM_SpooledFileMetricDefinition to related managed element
CIM_ManagedElement REF **Antecedent**	The CIM_ManagedElement that can have metrics of this type associated with it.	

IBM_SpooledFileMetricForME

Description: This association links a ManagedElement to the metric values being maintained for it.

Hierarchy: IBM_MetricForME, CIM_MetricForME

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A metric value for the ManagedElement.	IBM_SpooledFileMetricValue to related managed element
CIM_ManagedElement REF **Antecedent**	ManagedElement to which the metric values belong.	

IBM i System Metrics Providers

These providers provide capabilities to retrieve system performance data on IBM i.

The system metrics providers provide the capabilities to retrieve system performance data on IBM i. These providers use Collection Services data to track the elements of system performance of specific interest to you. You can use these providers to get system performance data of specific system resources (e.g. cpu, storage, job and network etc.).

Table 195. IBM i System 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 • **DiskSpaceUtilization** - Percentage of disk capacity actually used, id: MDDS40 • **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 • **DiskArmNumber** - Disk arm number, id:MDDS41 • **DiskDriveType** - Disk Drive Type, id:MDDS42 • **DiskMirrorStatus** - The status of local mirroring, id:MDDS43 • **DiskOperationalStatus** - Whether the disk is operational or not, id:MDDS44
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

Table 195. IBM i System Metrics (continued)

Resource/Base CIM Class	Metric
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 • **NetworkPortUtilizationPercentage** - Percentage of capacity actually used. This may be only a rough estimate, like bandwidth actually transferred vs. theoretical bandwidth limit, id: MDNP04
IBMOS400_OperatingSystem	• **AverageDiskActiveTimePercentage** - Average disk active time percentage, id: MDOS48 • **AverageDiskSpaceUtilization** - Average disk space utilization, id: MDOS4A • **AverageInteractiveJobsTransactionRate** - Average transaction rate of interactive jobs, id: MDOS42 • **AverageInteractiveResponseTime** - Average transaction response time of interactive jobs, id: MDOS43 • **AverageJobsTransactionRate** - Average transaction rate of jobs, id: MDOS41 • **AverageUserPoolFaultRate** - Average User Pool faults, id: MDOS46 • **BatchJobsLogicalDatabaseIORate** - Database IO operations rate of all batch jobs, id: MDOS45 • **CPUConsumptionIndex** - CPU time used divided by CPU time that might have been used by this operating system, id: MDOS25 • **ExternalViewKernelModePercentage** - IBM i always returns 0, id: MDOS2C • **ExternalViewTotalCPUPercentage** - External view CPU percentage, id: MDOS2A • **ExternalViewUserModePercentage** - External view user mode percentage, id: MDOS2B • **FreePhysicalMemory** - IBM i always returns 0, id: MDOS04 • **FreeSpaceInPagingFiles** - Free space in system ASP, id: MDOS05 • **FreeVirtualMemory** - Free space in system auxiliary storage pool (ASP), id: MDOS03 • **InteractiveJobsCPUPercentage** - Percentage CPU used for interactive jobs, id: MDOS40 • **InternalViewIdlePercentage** - Idle percentage as seen from within the operating system, id: MDOS24 • **InternalViewKernelModePercentage** - IBM i always returns 0, id: MDOS21 • **InternalViewTotalCPUPercentage** - User mode percentage as seen from within the operating system, id: MDOS23 • **InternalViewUserModePercentage** - IBM i always returns 0, id: MDOS22 • **KernelModeTime** -IBM i always returns 0, id: MDOS09 • **MaximumDiskActiveTimePercentage** - Maximum disk active time percentage, id: MDOS49 • **MaximumDiskSpaceUtilization** - Maximum disk space utilization, id: MDOS4B • **MaximumInteractiveResponseTime** - Maximum transaction response time of interactive jobs, id: MDOS44 • **MaximumUserPoolFaultRate** - Maximum User Pool faults, id: MDOS47 • **NumberOfProcesses** - Number of jobs active during the sample interval, id: MDOS02 • **NumberOfUsers** - Number of user sessions for which the operating system is currently storing state information, id: MDOS01 • **OperationalStatus** - IBM i 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 195. IBM i System Metrics (continued)

Resource/Base CIM Class	Metric
IBMOS400_Process	• **AccumulatedKernelModeTime** - IBM i 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 • **ActiveThreads** - Total number of active threads within the job, id: MDPR43 • **ExternalViewTotalCPUPercentage** - External view total CPU percentage, id: MDPR26 • **ExternalViewKernelModePercentage** - IBM i always returns 0, id: MDPR28 • **ExternalViewUserModePercentage** - External view user mode percentage, id: MDPR27 • **InteractiveResponseTime** - Averager response time of each transaction within job, id: MDPR41 • **InternalViewKernelModePercentage** - IBM i 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** - IBM i always returns 0, id: MDPR01 • **LogicalDatabaseIORate** - Number of database I/O operations per second, id: MDPR42 • **TransactionRate** - Number of transactions per second within job, id: MDPR40 • **TotalCPUTime** - Same as UserModeTime, id: MDPR03 • **TotalThreads** - Total number of threads within the job, id: MDPR44 • **UserModeTime** - The CPU time used by the JOB (including all secondary threads), id: MDPR02 • **PrimaryCommitOperationsPerformed** - Primary commit operations performed, id: MDPR45 • **PrimaryRollbackOperationsPerformed** - Primary rollback operations performed, id: MDPR46 • **NumberOfSeizesHeldByThread** - The number of seizes held by the thread, id: MDPR50 • **ProcessScopedLocksHeld** - The number of process scoped locks held by the thread, id: MDPR51 • **ThreadScopedLocksHeld** - The number of thread scoped locks held by the thread, id: MDPR52 • **ProcessScopedRecordLocksHeld** - The number of process scoped database record locks held by the thread, id: MDPR53 • **ThreadScopedRecordLocksHeld** - The number of thread scoped database record locks held by the thread, id: MDPR54
IBMOS400_VirtualProcessor	• **TotalCPUTimePercentage** - The time a virtual processor was used as a percentage of the elapsed interval time, id: MDPC01
IBM_StoragePool	• **ActiveToIneligibleTransitions** - Total number of transitions by processes assigned to the pool from active state to ineligible state, id: MDMP47 • **ActiveToWaitTransitions** - Total number of transitions by processes assigned to the pool from active state to wait state, id: MDMP45 • **DatabaseFaults** - Total number of pool database faults, id: MDMP41 • **DatabasePages** - Total number of pages of database data transferred from auxiliary storage to the pool, id: MDMP43 • **FaultRate** - Number of faults per second for the associated pool, id: MDMP40 • **NonDatabaseFaults** - Total number of pool nondatabase faults, id: MDMP42 • **NonDatabasePages** - Total number of pages of nondatabase data transferred from auxiliary storage to the pool, id: MDMP44 • **WaitToIneligibleTransitions** - Total number of transitions by processes assigned to the pool from wait state to ineligible state, id: MDMP46 • **ActivityLevel** - Maximum active threads, id: MDMP48

Table 195. IBM i System Metrics (continued)

Resource/Base CIM Class	Metric
IBMi_StorageController	• **DiskIOACPUUtilization** - Disk IOA CPU Utilization Percentage, id: MDDA01 • **DiskIOAAverageResponseTime** - Disk IOA Average Response Time, id: MDDA02 • **DiskIOAPercentDiskBusy** - Disk IOA Percent Disk Busy, id: MDDA03 • **DiskIOAReadsPerSecond** - Disk IOA Reads Per Second, id: MDDA04 • **DiskIOAWritesPerSecond** - Disk IOA Writes Per Second, id: MDDA05 • **DiskIOAOperationsPerSecond** - Disk IOA Operations Per Second, id: MDDA06 • **DiskIOAAverageKilobytesrPerRead** - Disk IOA Average Number of Kilobytes Per Read Operation, id: MDDA07 • **DiskIOAAverageKilobytesrPerWrite** - Disk IOA Average Number of Kilobytes Per Write Operation, id: MDDA08 • **DiskIOAAverageServiceTime** - Disk IOA Average Service Time, id: MDDA09 • **DiskIOAAverageWaitTime** - Disk IOA Average Wait Time, id: MDDA0A • **DiskIOATotalMegabytesRead** - Disk IOA Total Megabytes Read, id: MDDA0B • **DiskIOATotalMegabytesWritten** - Disk IOA Total Megabytes Written, id: MDDA0C

IBM_ColSrvMetricServiceConformsToBaseMetricsProfile

Description: IBM_ElementConformsToBaseMetricsProfile association defines the RegisteredProfiles to which the referenced ManagedElement is conformant. This association may apply to any Managed Element. Typical usage will apply it to a higher level instance, such as a System, NameSpace, or Service. When applied to a higher level instance, all constituent parts MUST behave appropriately in support of the ManagedElement's conformance to the named RegisteredProfile.

Hierarchy: IBM_MetricServiceConformsToBaseMetricsProfile, CIM_ElementConformsToProfile

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Note: There are some limitations for cross-namespace association employed by Pegasus Server (See Open Pegasus PEP#139 -- Cross Namespace Associations). Operation can be navigated only in a unidirectional way. And the association was only registered to namespace root/PG_InterOp, so associators/associatornames/references/referencenames from IBM_MetricService are not supporteded.

Property Name	Property Value (& data source)	Instance Mapping Rule
CIM_ManagedElement REF **ManagedElement**	The ManagedElement that conforms to the RegisteredProfile.	RegisteredProfile “Base Metrics Profile” associates to ColSrvMetricService instance.
CIM_RegisteredProfile REF **ConformantStandard**	The RegisteredBaseMetricsProfile to which the MetricService conforms to.	

IBM_ColSrvMetricServiceCapabilities

Description: Capabilities of a CIM_MetricService.

Hierarchy: IBM_MetricServiceCapabilities, CIM_MetricServiceCapabilities, CIM_EnabledLogicalElementCapabilities, CIM_Capabilities, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
uint16 **MetricsControlTypes**[]	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_BaseMetricDefinition identified by the value at the same array index in the ControllableMetrics property.	0 - Unknown

Property Name	class Defining Property	Property Description	Value or Value Location
uint16 **ManagedElementControlTypes**[]	CIM_MetricServiceCapabilities	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_ManagedElement identified by the value at the same array index in the ControllableManagedElements property.	0 - Unknown
boolean **ElementNameEditSupported**	CIM_EnabledLogicalElementCapabilities	Boolean indicating whether the ElementName can be modified.	FALSE
string **InstanceID**	CIM_Capabilities	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	"ColSrvMetricServiceCapabilities"
string **ElementName**	CIM_Capabilities	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	"Metric Service Capabilities"
uint16 SupportedMethods[]	CIM_MetricServiceCapabilities	Each enumeration corresponds to support for the like-named method of the MetricService.	7 - ControlSampleTimes

IBM_ColSrvMetricService

Description: The MetricService provides the ability to manage metrics.

Hierarchy: IBM_MetricService, CIM_MetricService, CIM_Service, CIM_EnabledLogicalElement, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, invokeMethod

Property Name	class Defining Property	Property Description	Value or Value Location
string **SystemCreationClassName**	CIM_Service	The CreationClassName of the scoping System.	"IBMOS400_ComputerSystem"
string **SystemName**	CIM_Service	The Name of the scoping System.	system name
string **CreationClassName**	CIM_Service	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	"IBM_ColSrvMetricService"
string **Name**	CIM_Service	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	"ColSrvMetricService"
boolean **Started**	CIM_Service	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	State of Performance Collection Service

Methods List:

Method Name	Class Defining Method	Description	Value or Value Location
ControlSampleTimes	CIM_MetricService	Method used to allow specification of the point in time metric gathering is to be started and to specify the preferred sample interval time for periodic data gathering. Whenever sampling for additional metrics is started, the settings specified by this method may be used.	

The method prototype:

string ControlSampleTimes(datetime PreferredSampleInterval, boolean RestartGathering, datetime StartSampleTime);

Input Parameter:

PreferredSampleInterval: Preferred sample interval time.

RestartGathering: Boolean that when set to TRUE requests that gathering of all metrics associated to the metric service is re-started with this method call.

StartSampleTime: Point in time when sampling for the metrics is to be started.

Return Code:

If 0 is returned, then the task completed successfully. Any other return code indicates an error condition.

0: Completed with No Error

1: Not Supported

2: Failed

IBM_ColSrvMetricServiceElementCapabilities

Description: IBM_MetricServiceElementCapabilities represents the association between ManagedElements and their Capabilities. The cardinality of the ManagedElement reference is Min(1), Max(1). This cardinality mandates the instantiation of the ElementCapabilities association for the referenced instance of Capabilities. ElementCapabilities describes the existence requirements and context for the referenced instance of ManagedElement. Specifically, the ManagedElement MUST exist and provides the context for the Capabilities.

Hierarchy: IBM_MetricServiceElementCapabilities,CIM_ElementCapabilities

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The managed element	ColSrvMetricService to ColSrvMetricServiceCapabilities
CIM_Capabilities REF **Capabilities**	The Capabilities object associated with the element.	

IBM_HostedColSrvMetricService

Description: IBM_HostedMetricService is an association between a Service and the System on which the functionality is located.

Hierarchy: IBM_HostedMetricService, CIM_HostedService, CIM_HostedDependency, CIM_Dependency

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property Value (and data source)	Instance mapping rule
CIM_Service REF **Dependent**	The Service hosted on the System.	Host System to ColSrvMetricService.
CIM_System REF **Antecedent**	The hosting System	

IBM_ColSrvMetricServiceAffectsElement

Description: To represents an association between a Service and the ManagedElements that might be affected by its execution.

Hierarchy: IBM_MetricServiceAffectsElement,CIM_ServiceAffectsElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF **AffectedElement**	The Managed Element that is affected by the Service	IBMOS400_ColSrvMetricDefinition to IBM_ColSrvMetricService.
CIM_Service REF **AffectingElement**	The Service that is affecting the ManagedElement.	
uint16 **ElementEffects**[]	An enumeration that describes the effect on the ManagedElement. This array corresponds to the OtherElementEffectsDescriptions array, where the latter provides details that are related to the high-level effects enumerated by this property.	5:Manages

IBMOS400_ColSrvMetricDefinition

Description: An IBMOS400_ColSrvMetricDefinition instance represents the definition aspects of a metric. The purpose of IBMOS400_ColSrvMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. IBMOS400_ColSrvMetricDefinition instance represents the definition aspects of a metric on IBM i.

Hierarchy: IBM_MetricDefinition,CIM_BaseMetricDefinition, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	class Defining Property	Property Description	Value or Value Location
string **Id**	CIM_BaseMetricDefinition	A string that uniquely identifies the metric definition. The use of OSF UUID/GUIDs is recommended.	Metric definition Id
string **Name**	CIM_BaseMetricDefinition	The name of the metric. This name does not have to be unique, but should be descriptive and may contain blanks.	Metric name
uint16 **DataType**	CIM_BaseMetricDefinition	The data type of the metric.	Metric data type
uint16 **Calculable**	CIM_BaseMetricDefinition	An enumerated value that describes the characteristics of the metric, for purposes of performing calculations	Metric calculable
string **Units**	CIM_BaseMetricDefinition	identifies the specific units of a value, like Bytes or Packets	Metric units
boolean **IsContinuous**	CIM_BaseMetricDefinition	IsContinuous indicates whether or not the metric value is continuous or scalar. Performance metrics are an example of a linear metric	Is Metric continuous
uint16 **ChangeType**	CIM_BaseMetricDefinition	ChangeType indicates how the metric value changes, in the form of typical combinations of finer grain attributes such as direction change, minimum and maximum values, and wrapping semantics	Metric change type
uint16 **TimeScope**	CIM_BaseMetricDefinition	TimeScope indicates the time scope to which the metric value applies.	Metric time scope
uint16 **GatheringType**	CIM_BaseMetricDefinition	GatheringType indicates how the metric values are gathered by the underlying instrumentation. This allows the client application to choose the right metric for the purpose.	Metric gathering type

Property Name	class Defining Property	Property Description	Value or Value Location
string **ElementName**	CIM_ManagedElement	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	Metric element name
string **Caption**	CIM_ManagedElement	The Caption property is a short textual description (one- line string) of the object.	Metric caption
string **Description**	CIM_ManagedElement	The Description property provides a textual description of the object.	Metric description

IBMOS400_ColSrvMetricValue

Description: IBMOS400_ColSrvMetricValue represents a metric value.

Hierarchy: IBM_MetricValue,CIM_BaseMetricValue, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	Class Defining Property	Property Description	Value or Value Location
string **InstanceID**	CIM_BaseMetricValue	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	This property of a metric value class must be generated and be unique for every instance (in time, duration, resource, and metric). See **NOTE 1**
string **MetricDefinitionId**	CIM_BaseMetricValue	The key of the BaseMetricDefinition instance for this CIM_BaseMetricValue instance value.	key defined for metric definition class
string **MeasuredElementName**	CIM_BaseMetricValue	A descriptive name for the element to which the metric value belongs (i.e., the measured element).	A short descriptive name for the managed element being measured as determined by the base class support. See **NOTE 2**
datetime **TimeStamp**	CIM_BaseMetricValue	Identifies the time when the value of a metric instance is computed. Note that this is different from the time when the instance is created.	The date and time of the sample interval.
datetime **Duration**	CIM_BaseMetricValue	Property that represents the time duration over which this metric value is valid.	Interval seconds from our DB file except for point metrics which will be returned as zero
String **MetricValue**	CIM_BaseMetricValue	The value of the metric represented as a string. Its original data type is specified in CIM_BaseMetricDefinition.	Metric value
boolean **Volatile**	CIM_BaseMetricValue	If true, Volatile indicates that the value for the next point in time may use the same object and just change its properties (such as the value or timestamp). If false, the existing objects remain unchanged and a new object is created for the new point in time.	TRUE

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex

Bytes	Description
6	Metric identifier (same as metric definition)
depends	Resource identifier : • ComputerSystem - none • OperatingSystem - none • Processor - CPU index • Process - qualified job name (job, user, nbr) • Network Port - Protocol + Line Description Name • Storage - Disk Serial Number • MemoryPool - Memory Pool Id • StorageController - Device Id

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- ComputerSystem: System name
- OperatingSystem: System name
- Process: Qualified Job name formatted as job-name+"/"+user-name +"/"+job_nbr
- Processor: "CPU "+ CPU index
- NetworkPort: Line description name.
- Storagae: :"Disk Unit " + 4 character logical disk unit number (data from QAMPDISK DB file) + [" "+ mirrored flag(the value is 'A' or 'B' from filed "DMFLAG" of QAMPDISK DB file)] (participates in case of mirrored disk) e.g. "Disk Unit 0001 A" or "Disk Unit 0002"
- MemoryPool: Pool Name
- StorageController: Device Id

IBMOS400_ColSrvMetricInstance

Description: Weak association of metric value objects with their metric definition. This association ties an instance of CIM_BaseMetricValue to its CIM_BaseMetricDefinition; it is weak because the existence of a CIM_BaseMetricValue depends on the existence of its associated CIM_BaseMetricDefinition: An instance of CIM_BaseMetricValue must not exist without being associated to its corresponding CIM_BaseMetricDefinition.

Hierarchy: IBM_MetricInstance, CIM_MetricInstance

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A CIM_BaseMetricValue instance holding the Value.	IBMOS400_ColSrvMetricDefinition to IBMOS400_ColSrvMetricValue
CIM_BaseMetricDefinition REF **Antecedent**	The CIM_BaseMetricDefinition for this particular CIM_BaseMetricValue.	

IBMOS400_ColSrvMetricDefForME

Description: This association ties a CIM_BaseMetricDefinition to a CIM_ManagedElement to define metrics for the latter. The metrics definition is given context by the ManagedElement, which is why the definition is dependent on the element.

Hierarchy: IBM_MetricDefForME,CIM_MetricDefForMe

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricDefinition REF **Dependent**	A CIM_BaseMetricDefinition for a CIM_ManagedElement.	IBMOS400_ColSrvMetricDefinition to related managed element
CIM_ManagedElement REF **Antecedent**	The CIM_ManagedElement that can have metrics of this type associated with it.	

IBMOS400_ColSrvMetricForME

Description: This association links a ManagedElement to the metric values being maintained for it.

Hierarchy: IBM_MetricForME, CIM_MetricForME

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A metric value for the ManagedElement.	IBMOS400_ColSrvMetricValue to related managed element
CIM_ManagedElement REF **Antecedent**	ManagedElement to which the metric values belong.	

IBM i Real Time Metrics Providers

These providers provide capabilities to retrieve real time system performance data on IBM i.

The real time metrics providers provide the capabilities to retrieve real time system performance data on IBM i. These providers use IBM i system API to retrieve real time performance data of system elements that specific interest to you. You can use these providers to get system real time performance data of specific system resources (e.g. operating system, job queue, subsystem and job etc.).

Table 196. IBM i System Real Time Metrics

Resource/Base CIM Class	Metric
IBMOS400_OperatingSystem	• **OSCurrentTemporaryStorageUsed** - OS current temporary storage used, id: MDOS4C. • **OSMaximumTemporaryStorageUsed** - OS maximum temporary storage used, id: MDOS4D. • **UncappedCPUCapacity** - Uncapped CPU capacity used, id: MDOS50. • **UsersCurrentlySignedOn** - Users Currently Signed On, id: MDOS51. • **UsersTemporarilyDisconnected** - The number of jobs that have been disconnected, id: MDOS52. • **JobsInSystem** - The total number of user jobs and system jobs that are currently in the system, id: MDOS53. • **BatchJobsWaitRun** - The number of batch jobs on the system that are currently waiting to run, including those that were submitted to run at a future date and time. Jobs on the job schedule that have not been submitted are not included, id: MDOS54. • **BatchJobsWaitMsg** - The number of batch jobs waiting for a reply to a message before they can continue to run, id: MDOS55. • **BatchJobsHeldOnQueue** - The number of batch jobs that were submitted, but were held before they could begin running, id: MDOS56. • **BatchJobsHeldWhileRunning** - The number of batch jobs that had started running, but are now held, id: MDOS57. • **BatchJobsOnHeldQueue** - The number of batch jobs on job queues that have been assigned to a subsystem, but are being held, id: MDOS58. • **BatchJobsRunning** - The number of batch jobs currently running on the system, id: MDOS59. • **BatchJobsEndWithLogs** - Batch jobs ended with printer output waiting to print, id: MDOS60. • **PermanentAddressesUsed** - The percentage of the maximum possible addresses for permanent objects that have been used, id: MDOS61. • **TemporaryAddressesUsed** - The percentage of the maximum possible addresses for temporary objects that have been used, id: MDOS62. • **BatchJobsEnding** - The number of batch jobs that are in the process of ending, id: MDOS63. • **BatchJobsOnUnassignedJobQueue** - The number of batch jobs on job queues that have not been assigned to a subsystem, id: MDOS64. • **UserSignedOffOutputWaiting** - The number of sessions that have ended with printer output files waiting to print, id: MDOS65. • **UserSuspendedByGroupJobs** - The number of user jobs that have been temporarily suspended by group jobs so that another job may be run, id: MDOS66. • **UserSuspendedBySysRequest** - The number of user jobs that have been temporarily suspended by system request jobs so that another job may be run, id: MDOS67.
IBMOS400_Process	• **MaximumProcessingUnitTimeAllowed** - Maximum processing unit time allowed, id: MDPR47. • **JobTemporaryStorageUsed** - Job's temporary storage used, id: MDPR48. • **JobMaximumTemporaryStorageAllowed** - Job's maximum temporary storage allowed, id: MDPR49. • **JobMaximumTemporaryStorageUsed** - Job's maximum temporary storage used, id: MDPR4A.
IBM_JobQueue	• **JobQueueStatus** - The status of the job queue. The status may be one of RELEASED & HELD, id: MDJQ01. • **JobsNumberInJobQueue** - The number of jobs in the queue, id: MDJQ02.
IBM_Subsystem	• **PercentMaximumJobs** - Current active jobs % maximum active jobs in the subsystem, id: MDSS01. • **ActiveJobsInSubsystem** - The number of active jobs in the subsystem, id: MDSS02. • **SubsystemStatus** - The status of the Subsystem. The status may be one of *ACTIVE & *INACTIVE, id: MDSS03.

Table 196. IBM i System Real Time Metrics (continued)

Resource/Base CIM Class	Metric
IBM_AdminDomain	• **AverageResponseTime** - The average server response time in milliseconds , id:MDNS01. • **PasswordViolations** - The number of server password violations , id:MDNS02. • **SessionStartsPerMinute** - The number of server session starts per minute, id:MDNS03. • **FileOpensPerMinute** - The number of file opens for the whole server per minute, id: MDNS04. • **KbytesReceivedPerMinute** - The number of server kilo-bytes received from the network, id: MDNS05. • **KbytesSentPerMinute** - The number of server kilo-bytes sent to the network, id: MDNS06. • **PrintJobsQueuedPerMinute** - The number of server print jobs spooled, id: MDNS07.
IBM_IPProtocolEndpoint	• **IPInterfaceStatus** - The current status of the logic interface, id: MDNI01. • **IPInterfaceAutoStartStatus** - Whether the IP interface is started automatically, id: MDNI02.

IBM_RealTimeMetricServiceConformsToBaseMetricsProfile

Description: IBM_ElementConformsToBaseMetricsProfile association defines the RegisteredProfiles to which the referenced ManagedElement is conformant. This association may apply to any Managed Element. Typical usage will apply it to a higher level instance, such as a System, NameSpace, or Service. When applied to a higher level instance, all constituent parts MUST behave appropriately in support of the ManagedElement's conformance to the named RegisteredProfile.

Hierarchy: IBM_MetricServiceConformsToBaseMetricsProfile, CIM_ElementConformsToProfile

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Note: There are some limitations for cross-namespace association employed by Pegasus Server (See Open Pegasus PEP#139 -- Cross Namespace Associations). Operation can be navigated only in a unidirectional way. And the association was only registered to namespace root/PG_InterOp, so associators/associatornames/references/referencenames from IBM_MetricService are not supporteded.

Property Name	Property Value (& data source)	Instance Mapping Rule
CIM_ManagedElement REF **ManagedElement**	The ManagedElement that conforms to the RegisteredProfile.	This should be 1 to 1 association between IBM_RegisteredBaseMetricsProfile and IBM_RealTimeMetricService.
CIM_RegisteredProfile REF **ConformantStandard**	The RegisteredBaseMetricsProfile to which the MetricService conforms to.	

IBM_RealTimeMetricServiceCapabilities

Description: Capabilities of a CIM_MetricService.

Hierarchy: IBM_MetricServiceCapabilities, CIM_MetricServiceCapabilities, CIM_EnabledLogicalElementCapabilities, CIM_Capabilities, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	Property Description	Value or Value Location
uint16 **MetricsControlTypes**[]	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_BaseMetricDefinition identified by the value at the same array index in the ControllableMetrics property.	0 (Unknown)
uint16 **ManagedElementControlTypes**[]	This property identifies the type of control supported by the associated CIM_MetricService instance for the CIM_ManagedElement identified by the value at the same array index in the ControllableManagedElements property.	0 (Unknown)
boolean **ElementNameEditSupported**	Boolean indicating whether the ElementName can be modified.	FALSE

Property Name	Property Description	Value or Value Location
string **InstanceID** (key)	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	RealTimeMetricServiceCapabilities
string **ElementName**	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	Metric Service Capabilities
string **SupportedMethods**[]	Each enumeration corresponds to support for the like-named method of the MetricService.	7 (ControlSampleTimes)

IBM_RealTimeMetricService

Description: The MetricService provides the ability to manage metrics.

Hierarchy: IBM_MetricService, CIM_MetricService, CIM_Service, CIM_EnabledLogicalElement, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, invokeMethod

Property Name	Property Description	Value or Value Location
string **SystemCreationClassName**	The CreationClassName of the scoping System.	IBMOS400_ComputerSystem
string **SystemName**	The Name of the scoping System.	system name
string **CreationClassName**	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	IBM_RealTimeMetricService
string **Name**	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	RealTimeMetricService
boolean **Started**	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	TRUE
sint32 **SamplingInterval**	User setting preferred sampling interval with second unit.	See **NOTE 1**

Note 1: This value is used to control indication generating interval, which can be modified by user through ControlSamplingTime method. For performance consideration, the value should greater than or equal to 10(s). The minimum threshold of this property is defined according to two aspects: number of real time metric indications that can be generated in sampling interval and CPU time consuming. Current " IBM® Systems Director Monitors" retrieving real time metric value mechanism also has an at least 10(s) interval.

Methods List:

Method Name	Description	Value or Value Location
ControlSampleTimes	Method used to allow specification of the point in time metric gathering is to be started and to specify the preferred sample interval time for periodic data gathering. Whenever sampling for additional metrics is started, the settings specified by this method may be used.	

The method prototype:

string ControlSampleTimes(datetime PreferredSampleInterval, boolean RestartGathering, datetime StartSampleTime);

Input Parameter:

PreferredSampleInterval: Preferred sample interval time.

RestartGathering: Boolean that when set to TRUE requests that gathering of all metrics associated to the metric service is re-started with this method call.

StartSampleTime: Point in time when sampling for the metrics is to be started.

Return Code:

If 0 is returned, then the task completed successfully. Any other return code indicates an error condition.

0: Completed with No Error

1: Not Supported

2: Failed

Performance Limitation:

Considering realtime metric indication mechanism performance, IBM_RealTimeMetricService request user to provide PreferredSampleInterval parameter value must greater than or equal to 10000(ms). Otherwise, the method reutrns 1 (Not Supported).

IBM_RealTimeMetricServiceElementCapabilities

Description: IBM_MetricServiceElementCapabilities represents the association between ManagedElements and their Capabilities. The cardinality of the ManagedElement reference is Min(1), Max(1). This cardinality mandates the instantiation of the ElementCapabilities association for the referenced instance of Capabilities. ElementCapabilities describes the existence requirements and context for the referenced instance of ManagedElement. Specifically, the ManagedElement MUST exist and provides the context for the Capabilities.

Hierarchy: IBM_MetricServiceElementCapabilities,CIM_ElementCapabilities

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value (and data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The managed element	This should be 1 to 1 association between IBM_RealTimeMetricService and IBM_RealTimeMetricServiceCapabilities.
CIM_Capabilities REF **Capabilities**	The Capabilities object associated with the element.	

IBM_HostedRealTimeMetricService

Description: IBM_HostedMetricService is an association between a Service and the System on which the functionality is located.

Hierarchy: IBM_HostedMetricService, CIM_HostedService, CIM_HostedDependency, CIM_Dependency

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property Value (and data source)	Instance mapping rule
CIM_Service REF **Dependent**	The Service hosted on the System.	This should be 1 to 1 association between IBMOS400_ComputerSystem and IBM_RealTimeMetricService.
CIM_System REF **Antecedent**	The hosting System	

IBM_SystemStatusMetricServiceAffectsElement

Description: To represents an association between a Service and the ManagedElements that might be affected by its execution.

Hierarchy: IBM_MetricServiceAffectsElement,CIM_ServiceAffectsElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF **AffectedElement**	The Managed Element that is affected by the Service	This should be 1 to N association between IBM_RealTimeMetricService and IBM_SystemStatusMetricDefinition
CIM_Service REF **AffectingElement**	The Service that is affecting the ManagedElement.	
uint16 **ElementEffects**[]	An enumeration that describes the effect on the ManagedElement. This array corresponds to the OtherElementEffectsDescriptions array, where the latter provides details that are related to the high-level effects enumerated by this property.	5 (Manages)

IBM_JobQueueMetricServiceAffectsElement

The class definition is the same as IBM_SystemStatusMetricServiceAffectsElement. It represents the association between IBM_RealTimeMetricService and IBM_JobQueueMetricDefinition.

IBM_JobMetricServiceAffectsElement

The class definition is the same as IBM_SystemStatusMetricServiceAffectsElement. It represents the association between IBM_RealTimeMetricService and IBM_JobMetricDefinition.

IBM_SubsystemMetricServiceAffectsElement

The class definition is the same as IBM_SystemStatusMetricServiceAffectsElement. It represents the association between IBM_RealTimeMetricService and IBM_SubstytemMetricDefinition.

IBM_NetServerMetricServiceAffectsElement

The class definition is the same as IBM_SystemStatusMetricServiceAffectsElement. It represents the association between IBM_RealTimeMetricService and IBM_NetServerMetricDefinition.

IBM_NetWorkMetricServiceAffectsElement

The class definition is the same as IBM_SystemStatusMetricServiceAffectsElement. It represents the association between IBM_RealTimeMetricService and IBM_NetWorkMetricDefinition.

IBM_SystemStatusMetricDefinition

Description: An IBM_SystemStatusMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_SystemStatusMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class.

Hierarchy: IBM_MetricDefinition,CIM_BaseMetricDefinition, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	Property Description	Value or Value Location
string **Id** (key)	A string that uniquely identifies the metric definition. The use of OSF UUID/GUIDs is recommended.	Metric definition Id
string **Name**	The name of the metric. This name does not have to be unique, but should be descriptive and may contain blanks.	Metric name
uint16 **DataType**	The data type of the metric.	Metric data type
uint16 **Calculable**	An enumerated value that describes the characteristics of the metric, for purposes of performing calculations	Metric calculable
string **Units**	identifies the specific units of a value, like Bytes or Packets	Metric units
boolean **IsContinuous**	IsContinuous indicates whether or not the metric value is continuous or scalar. Performance metrics are an example of a linear metric	Is Metric continuous
uint16 **ChangeType**	ChangeType indicates how the metric value changes, in the form of typical combinations of finer grain attributes such as direction change, minimum and maximum values, and wrapping semantics	Metric change type
uint16 **TimeScope**	TimeScope indicates the time scope to which the metric value applies.	Metric time scope
uint16 **GatheringType**	GatheringType indicates how the metric values are gathered by the underlying instrumentation. This allows the client application to choose the right metric for the purpose.	Metric gathering type

Property Name	Property Description	Value or Value Location
string **ElementName**	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. (Note: Name does not have to be unique within a namespace.)	Metric element name
string **Caption**	The Caption property is a short textual description (one- line string) of the object.	Metric caption
string **Description**	The Description property provides a textual description of the object.	Metric description

IBM_SystemStatusMetricValue

Description: IBM_SystemStatusMetricValue represents a metric value.

Hierarchy: IBM_MetricValue,CIM_BaseMetricValue, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property Name	Class Defining Property	Property Description	Value or Value Location
string **InstanceID** (key)	CIM_BaseMetricValue	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	This property of a metric value class must be generated and be unique for every instance (in time, duration, resource, and metric). See **NOTE 1**
string **MetricDefinitionId**	CIM_BaseMetricValue	The key of the BaseMetricDefinition instance for this CIM_BaseMetricValue instance value.	key defined for metric definition class
string **MeasuredElementName**	CIM_BaseMetricValue	A descriptive name for the element to which the metric value belongs (i.e., the measured element).	A short descriptive name for the managed element being measured as determined by the base class support. See **NOTE 2**
datetime **TimeStamp**	CIM_BaseMetricValue	Identifies the time when the value of a metric instance is computed. Note that this is different from the time when the instance is created.	The date and time of the sample interval.
datetime **Duration**	CIM_BaseMetricValue	Property that represents the time duration over which this metric value is valid.	0
String **MetricValue**	CIM_BaseMetricValue	The value of the metric represented as a string. Its original data type is specified in CIM_BaseMetricDefinition.	Metric value
boolean **Volatile**	CIM_BaseMetricValue	If true, Volatile indicates that the value for the next point in time may use the same object and just change its properties (such as the value or timestamp). If false, the existing objects remain unchanged and a new object is created for the new point in time.	TRUE

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	System Name

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- Operating System: System name

IBM_SystemStatusMetricInstance

Description: Weak association of metric value objects with their metric definition. This association ties an instance of CIM_BaseMetricValue to its CIM_BaseMetricDefinition; it is weak because the existence of a CIM_BaseMetricValue depends on the existence of its associated CIM_BaseMetricDefinition:
An instance of CIM_BaseMetricValue must not exist without being associated to its corresponding CIM_BaseMetricDefinition.

Hierarchy: IBM_MetricInstance, CIM_MetricInstance

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A CIM_BaseMetricValue instance holding the Value.	This should be N to N association between IBM_SystemStatusMetricDefinition and IBM_SystemStatusMetricValue.
CIM_BaseMetricDefinition REF **Antecedent**	The CIM_BaseMetricDefinition for this particular CIM_BaseMetricValue.	

IBM_SystemStatusMetricDefForME

Description: This association ties a CIM_BaseMetricDefinition to a CIM_ManagedElement to define metrics for the latter. The metrics definition is given context by the ManagedElement, which is why the definition is dependent on the element.

Hierarchy: IBM_MetricDefForME,CIM_MetricDefForMe

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricDefinition REF **Dependent**	A CIM_BaseMetricDefinition for a CIM_ManagedElement.	This should be 1 to N association between IBMOS400_OperatingSystem and IBM_SystemStatusMetricDefinition.
CIM_ManagedElement REF **Antecedent**	The CIM_ManagedElement that can have metrics of this type associated with it.	

IBM_SystemStatusMetricForME

Description: This association links a ManagedElement to the metric values being maintained for it.

Hierarchy: IBM_MetricForME, CIM_MetricForME

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Instance mapping rule
CIM_BaseMetricValue REF **Dependent**	A metric value for the ManagedElement.	This should be 1 to N association between IBMOS400_OperatingSystem and IBM_SystemStatusMetricValue.
CIM_ManagedElement REF **Antecedent**	ManagedElement to which the metric values belong.	

IBM_JobQueueMetricDefinition

An IBM_JobQueueMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_JobQueueMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. The class definition is the same as IBM_SystemStatusMetricDefinition.

IBM_JobQueueMetricValue

IBM_JobQueueMetricValue represents a metric value. The class definition is the same as IBM_SystemStatusMetricValue. The two notes about the class are listed below:

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Job Queue – qualified job queue name

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- Job Queue: Job Queue Library Name/Job Queue Name.

IBM_JobQueueMetricInstance

The class definition is the same as IBM_SystemStatusMetricInstance. It represents the association between IBM_JobQueueMetricDefinition and IBM_JobQueueMetricValue.

IBM_JobQueueMetricDefForME

The class definition is the same as IBM_SystemStatusMetricDefForME. It represents the association between IBM_JobQueueMetricDefinition and IBM_JobQueue.

IBM_JobQueueMetricForME

The class definition is the same as IBM_SystemStatusMetricForME. It represents the association between IBM_JobQueueMetricValue and IBM_JobQueue.

IBM_JobMetricDefinition

An IBM_JobMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_JobMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. The class definition is the same as IBM_SystemStatusMetricDefinition.

IBM_JobMetricValue

IBM_JobMetricValue represents a metric value. The class definition is the same as IBM_SystemStatusMetricValue. The two notes about the class are listed below:

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Job identify

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- Process: Job Name/User Name/Job Number.

IBM_JobMetricInstance

The class definition is the same as IBM_SystemStatusMetricInstance. It represents the association between IBM_JobMetricDefinition and IBM_JobMetricValue.

IBM_JobMetricDefForME

The class definition is the same as IBM_SystemStatusMetricDefForME. It represents the association between IBM_JobMetricDefinition and IBMOS400_Process.

IBM_JobMetricForME

The class definition is the same as IBM_SystemStatusMetricForME. It represents the association between IBM_JobMetricValue and IBMOS400_Process.

IBM_SubsystemMetricDefinition

An IBM_SubsystemMetricDefinition instance represents the definition aspects of a metric. The purpose of IBM_SubsystemMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. The class definition is the same as IBM_SystemStatusMetricDefinition.

IBM_SubsystemMetricValue

IBM_SubsystemMetricValue represents a metric value. The class definition is the same as IBM_SystemStatusMetricValue. The two notes about the class are listed below:

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Subsystem – qualified subsystem name

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- Subsystem: Subsystem Library Name/Subsystem Name.

IBM_SubsystemMetricInstance

The class definition is the same as IBM_SystemStatusMetricInstance. It represents the association between IBM_SubsystemMetricDefinition and IBM_SubsystemMetricValue.

IBM_SubsystemMetricDefForME

The class definition is the same as IBM_SystemStatusMetricDefForME. It represents the association between IBM_SubsystemMetricDefinition and IBM_Subsystem.

IBM_SubsystemMetricForME

The class definition is the same as IBM_SystemStatusMetricForME. It represents the association between IBM_SubsystemMetricValue and IBM_Subsystem.

IBM_NetServerMetricDefinition

An IBM_NetServerMetricDefinition instance represents the definition aspects of metrics related to NetServer. The purpose of IBM_NetServerMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. The class definition is the same as IBM_SystemStatusMetricDefinition.

IBM_NetServerMetricValue

IBM_NetServerMetricValue represents a metric value. The class definition is the same as IBM_SystemStatusMetricValue. The two notes about the class are listed below:

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Server Name

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- AdminDomain: Server Name.

IBM_NetServerMetricInstance

The class definition is the same as IBM_SystemStatusMetricInstance. It represents the association between IBM_NetServerMetricDefinition and IBM_NetServerMetricValue.

IBM_NetServerMetricDefForME

The class definition is the same as IBM_SystemStatusMetricDefForME. It represents the association between IBM_NetServerMetricDefinition and IBM_AdminDomain.

IBM_NetServerMetricForME

The class definition is the same as IBM_SystemStatusMetricForME. It represents the association between IBM_NetServerMetricValue and IBM_AdminDomain.

IBM_NetWorkMetricDefinition

An IBM_NetWorkMetricDefinition instance represents the definition aspects of metrics related to network interface. The purpose of IBM_NetWorkMetricDefinition is to provide a convenient mechanism for introducing all metrics definition at runtime and capturing its instance values in a separate class. The class definition is the same as IBM_SystemStatusMetricDefinition.

IBM_NetWorkMetricValue

IBM_NetWorkMetricValue represents a metric value. The class definition is the same as IBM_SystemStatusMetricValue. The two notes about the class are listed below:

NOTE 1: InstanceId must be generated and be unique for every instance (in time, duration, resource, and metric). It is a string of undefined length. In this release we still do not support the historical metrics. Currently, our pattern is as follows:

Bytes	Description
8	IPL Identifier (MATMATR 01F8), represented as printable hex
6	Metric identifier (same as metric definition)
depends	Resource Name (resource name of IBM_IPProtocolEndpoint instance), resource name is built up as: Name-LineName-IPVersionSupport

NOTE 2: MeasuredElementName is a short descriptive name for the managed element being measured as determined by the base class support:

- IBM_IPProtocolEndpoint: Resource Name, resource name is built up as Name-LineName-IPVersionSupport.

IBM_NetWorkMetricInstance

The class definition is the same as IBM_SystemStatusMetricInstance. It represents the association between IBM_NetWorkMetricDefinition and IBM_NetWorkMetricValue.

IBM_NetWorkMetricDefForME

The class definition is the same as IBM_SystemStatusMetricDefForME. It represents the association between IBM_NetWorkMetricDefinition and IBM_IPProtocolEndPoint.

IBM_NetWorkMetricForME

The class definition is the same as IBM_SystemStatusMetricForME. It represents the association between IBM_NetWorkMetricValue and IBM_IPProtocolEndPoint.

IBM i Metrics Value Modification Indication Providers

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.

The following CIM classes have been implemented as IBM-supplied providers to provide Metrics Value Modification Indication:

- IBM_ColSrvMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBMOS400_ColSrvMetricValue of system metrics.
- IBM_FileMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_FileMetricValue of file metrics.
- IBM_SpooledFileMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_SpooledFileMetricValue of spooled file metrics.
- IBM_SystemStatusMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_SystemStatusMetricValue of system status metrics.

- IBM_JobMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_JobMetricValue of job metrics.
- IBM_JobQueueMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_JobQueueMetricValue of job queue metrics.
- IBM_SubsystemMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_SubsystemMetricValue of real time system metrics.
- IBM_NetServerMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_NetServerMetricValue of real time system metrics.
- IBM_NetWorkMetricValueModification: a subclass of IBM_MetricValueModification that represents a modification event of IBM_NetWorkMetricValue of real time system metrics.

IBM_ColSrvMetricValueModification

Description: IBM_ColSrvMetricValueModification represents a modification event of IBMOS400_ColSrvMetricValue. CIM_InstModification notifies when the instance of IBMOS400_ColSrvMetricValue is changed. The client can get the previous information and current information of IBMOS400_ColSrvMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have **LOW PERFORMANCE**, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the CPU Utilization on the system, the query can be specified *"select I.** from *CIM_InstModification I* where *I.**SourceInstance*** isa *CIM_BaseMetricValue* and *I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDOS2A' "*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBMOS400_ColSrvMetricValue instance
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBMOS400_ColSrvMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBMOS400_ColSrvMetricValue
string **SourceInstanceHost**	CIM_InstIndication	The host name or IP address of the SourceInstance.	Host name

Property Name	Class Defining Property	Property Description	Value or Value Location
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_FileMetricValueModification

Description: IBM_FileMetricValueModification represents a modification event of IBM_FileMetricValue. CIM_InstModification notifies when the instance of IBM_FileMetricValue is changed. The client can get the previous information and current information of IBM_FileMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: To file metrics, the file name MUST be specified on the **MeasuredElementName** or the **InstanceId**. Otherwise, the filter will be regarded as an invalid filter. For example, if the client is interested in the size of file "/QSYS.LIB/QHST08295A.FILE" on the system, the query can be specified *"select I.** from *CIM_InstModification I* where *I.**SourceInstance*** isa *CIM_BaseMetricValue* and *I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDFL40'* and *I.SourceInstance.CIM_BaseMetricValue::**MeasuredElementName** = '/QSYS.LIB/QHST08295A.FILE '* ".

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_FileMetricValue instance
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_FileMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_FileMetricValue
string **SourceInstanceHost**	CIM_InstIndication	The host name or IP address of the SourceInstance.	Host name

Property Name	Class Defining Property	Property Description	Value or Value Location
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_SpooledFileMetricValueModification

Description: IBM_SpooledFileMetricValueModification represents a modification event of IBM_SpooledFileMetricValue. CIM_InstModification notifies when the instance of IBM_SpooledFileMetricValue is changed. The client can get the previous information and current information of IBM_SpooledFileMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the spooled files in ASP group, the query can be specified *“select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDSP40' ”*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_SpooledFileMetricValue instance
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_SpooledFileMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_SpooledFileMetricValue

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstanceHost**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_SystemStatusMetricValueModification

Description: IBM_SystemStatusMetricValueModification represents a modification event of IBM_SystemStatusMetricValue. CIM_InstModification notifies when the instance of IBM_SystemStatusMetricValue is changed. The client can get the previous information and current information of IBM_SystemStatusMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the total number of user jobs and system jobs that are currently in the system, the query can be specified *"select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDOS53' "*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_SystemStatusMetricValue instance.

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_SystemStatusMetricValue instance.
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_SystemStatusMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_JobMetricValueModification

Description: IBM_JobMetricValueModification represents a modification event of IBM_JobMetricValue. CIM_InstModification notifies when the instance of IBM_JobMetricValue is changed. The client can get the previous information and current information of IBM_JobMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the storage utilization of jobs, the query can be specified *"select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDPR48' "*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_JobMetricValue instance.

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_JobMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path.	Object path of IBM_JobMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_JobQueueMetricValueModification

Description: IBM_JobQueueMetricValueModification represents a modification event of IBM_JobQueueMetricValue. CIM_InstModification notifies when the instance of IBM_JobQueueMetricValue is changed. The client can get the previous information and current information of IBM_JobQueueMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the status of JobQueues, the query can be specified *“select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDJQ01' ”*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_JobQueueMetricValue instance

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_JobQueueMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_JobQueueMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_SubsystemMetricValueModification

Description: IBM_SubsystemMetricValueModification represents a modification event of IBM_SubsystemMetricValue. CIM_InstModification notifies when the instance of IBM_SubsystemMetricValue is changed. The client can get the previous information and current information of IBM_SubsystemMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the status of Subsystems, the query can be specified *"select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDSS03' "*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_SubsystemMetricValue instance

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_SubsystemMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_SubsystemMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_NetServerMetricValueModification

Description: IBM_NetServerMetricValueModification represents a modification event of IBM_NetServerMetricValue. CIM_InstModification notifies when the instance of IBM_NetServerMetricValue is changed. The client can get the previous information and current information of IBM_NetServerMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the number of server password violations of NetServer, the query can be specified *“select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDNS02' ”*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_NetServerMetricValue instance

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_NetServerMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path	Object path of IBM_NetServerMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

IBM_NetWorkMetricValueModification

Description: IBM_NetWorkMetricValueModification represents a modification event of IBM_NetWorkMetricValue. CIM_InstModification notifies when the instance of IBM_NetWorkMetricValue is changed. The client can get the previous information and current information of IBM_NetWorkMetricValue from the indication instance.

Hierarchy: IBM_MetricValueModification, CIM_InstIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

Important: Recommend the indication queries filter on either the **InstanceId** or the **MetricDefinitionId** property. The indication provider will have LOW PERFORMANCE, if empty filters or a filter that provides only properties other than these two. For example, if the client is interested in the status of the logical network interface, the query can be specified *“select I.* from CIM_InstModification I where **I.SourceInstance** isa CIM_BaseMetricValue and I.SourceInstance.CIM_BaseMetricValue::**MetricDefinitionId** = 'MDNI01' ”*

Property Name	Class Defining Property	Property Description	Value or Value Location
string **PreviousInstance**	CIM_InstModification	A copy of the 'previous' instance whose change generated the Indication. PreviousInstance contains 'older' values of an instance's properties (as compared to SourceInstance), selected by the IndicationFilter's Query.	Previous IBM_NetWorkMetricValue instance

Property Name	Class Defining Property	Property Description	Value or Value Location
string **SourceInstance**	CIM_InstIndication	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the Indication Filter's Query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	Current IBM_NetWorkMetricValue instance
string **SourceInstanceModelPath**	CIM_InstIndication	The Model Path of the SourceInstance. The following format MUST be used to encode the Model Path.	Object path of IBM_NetWorkMetricValue
string **SourceInstanceHost**	CIM_Indication	The host name or IP address of the SourceInstance.	Host name
string **IndicationIdentifier**	CIM_Indication	An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications (see the CorrelatedIndications array). Its value SHOULD be unique as long as correlations are reported, but MAY be reused or left NULL if no future Indications will reference it in their CorrelatedIndications array.	GUID
datetime **IndicationTime**	CIM_Indication	The time and date of creation of the Indication. The property may be set to NULL if the entity creating the Indication is not capable of determining this information. **Note:** IndicationTime may be the same for two Indications that are generated in rapid succession.	Current time of creating the Indication

Related information

The Open Group: CIMIndicationProvider documentation

IBM i File Monitor Providers

DMTF management profile, DSP 1002 Diagnostics Profile, is extended to support the file content monitor. The Diagnostics Profile extends the management capability of referencing profiles by adding the capability to run diagnostic services in a managed system. This profile includes a specification of the Diagnostic Test Service, its configuration, its associated capabilities, its logging mechanisms, and its profile registration information.

Following table lists all file monitor providers:

Provider Name	Implements CIM class	Provider Type
IBM_FileMonitorService	CIM_DignostcTest	Instance, Method
IBM_FileMonitorServiceCapabilities	CIM_DiagnosticServiceCapabilities	Instance
IBMi_FileContentMonitorSetting	CIM_DiagnosticSetting	Instance
IBM_FileContentMonitorJob	CIM_ConcreteJob	Instance, Method
IBM_HostedFileMonitorService	CIM_HostedService	Association
IBM_FileMonitorServiceElementCapabilities	CIM_ElementCapabilities	Association
IBM_OwningFileContentMonitorJob	CIM_OwningJobElement	Association

Provider Name	Implements CIM class	Provider Type
IBM_FileContentChangedIndication	CIM_AlertIndication	Indication

IBMi_FileContentMonitorSetting

Description: Specific file content monitor parameters and execution instructions are defined by this class. All instance of this class is created by user. And the lifecycle is also maintained by the user

Hierarchy: CIM_DiagnosticSetting, CIM_Setting, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, createInstance, modifyInstance, deteleInstance

Note: All instances of this class are created by client. And the lifecycle is also maintained by the client. If the instance will not be used any longer, it should be deleted manually.

Property Name	Class Defining Property	Property Description	Value or Value Location
string **MonitoredFileName**	IBMi_FileContentMonitorSetting	The absolute path of monitored file	Input by User
uint16 **FileCCSID**	IBMi_FileContentMonitorSetting	The file encoding CCSID	Input by User
string **MonitoredString**	IBMi_FileContentMonitorSetting	The string to be monitored. It could be a regular expression	Input by User
string **SettingID**	CIM_DiagnosticSetting	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	Input by User
string **ElementName**	CIM_ManagedElement	The user-friendly name for this instance of SettingData. In addition, the user-friendly name can be used as an index property for a search or query.	Input by User
string **Caption**	CIM_ManagedElement	The Caption property is a short textual description (one- line string) of the object.	Input by User
string **Description**	CIM_ManagedElement	The Description property provides a textual description of the object.	Defined by User

IBM_FileContentMonitorJob

Description: A concrete version of Job. This class represents a generic and instantiable unit of work, such as the asynchronous task.

Hierarchy: CIM_ConcreteJob, CIM_Job, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, deleteInstance, invokeMethod

Property Name	Class Defining Property	Property Description	Value or Value Location
Datetime **TimeOfLastStateChange**	CIM_ConcreteJob	The date or time when the state of the Job last changed	Date Time of create/modify the instance
uint16 **JobState**	CIM_ConcreteJob	JobState is an integer enumeration that indicates the operational state of a Job. It can also indicate transitions between these states, for example, 'Shutting Down' and 'Starting'. ValueMap ({"2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13..32767", "32768..65535"}), Values ({"New", "Starting", "Running", "Suspended", "Shutting Down", "Completed", "Terminated", "Killed", "Exception", "Service", "Query Pending", "DMTF Reserved", "Vendor Reserved"})	Current Job State
String **InstanceID**	CIM_ConcreteJob	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	GUID+”:”+ElementName
string **Name**	CIM_ConcreteJob	The user-friendly name for this instance of a Job.	"ElementeName
Boolean **DeleteOnCompletion**	CIM_Job	Indicates whether or not the job should be automatically deleted upon completion. **Note** that the 'completion' of a recurring job is defined by its JobRunTimes or UntilTime properties, or when the Job is terminated by manual intervention. If this property is set to false and the job completes, then the extrinsic method DeleteInstance must be used to delete the job instead of updating this property.	"Metric Service Capabilities"
Datetime **StartTime**	CIM_Job	The time that the Job was actually started.	Job Start Time
String **ElementName**	CIM_ManagedElement	A user-friendly name for the object. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information.	{IBMi_FileContentMonitorSetting. SettingID}

Methods List:

Method Name	Class Defining Method	Description	Value or Value Location
RequestStateChange	CIM_ConcreteJob	Requests that the state of the job be changed to the value specified in the RequestedState parameter. Invoking the RequestStateChange method multiple times could result in earlier requests being overwritten or lost. If 0 is returned, then the task completed successfully. Any other return code indicates an error condition.	This method is used to control the job, such as cancel the job etc. We support below state change: **Start (2)** - changes the state to \'Running\'. **Suspend (3)** - stops the job temporarily. The intention is to subsequently restart the job with "\'Start\'. It might be possible to enter the \'Service\' state while suspended. (This is job-specific.) **Terminate (4)** - stops the job cleanly, saving data, preserving the state, and shutting down all underlying processes in an orderly manner. **Kill (5)** - terminates the job immediately with no requirement to save data or preserve the state.

The method **`RequestStateChange`** requests that the state of the job be changed to the value specified in the RequestedState parameter. The method prototype:

uint32 **RequestStateChange**(uint16 RequestedState, datetime TimeoutPeriod);

Input Parameter:

RequestedState: the state that the job will be changed to. The allowed values of IBM_FileContentMonitorJob are 2, 3, 4 and 5.

TimeoutPeriod: A timeout period that specifies the maximum amount of time that the client expects the transition to the new state to take. The interval format must be used to specify the TimeoutPeriod. A value of 0 or a null parameter indicates that the client has no time requirements for the transition.

Return Code:

If 0 is returned, then the task completed successfully. Any other return code indicates an error condition.

0: Completed with No Error

2: Unknown/Unspecified Error

3: Can NOT complete within Timeout Period

4: Failed

5: Invalid Parameter

6: In Use

4096: Method Parameters Checked - Transition Started

4097: Invalid State Transition

4098: Use of Timeout Parameter Not Supporteded

4099: Busy

Note: Invoking the RequestStateChange method multiple times could result in earlier requests being overwritten or lost.

IBM_FileMonitorService

Description: Service to control the migration of virtual systems between host systems, including pre-checking to determine if an intended migration operation is likely to succeed.

Hierarchy: CIM_DiagnosticTest, CIM_Service, CIM_EnabledLogicalElement, CIM_LogicalElement, CIM_ManagedSystemElement, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, invokeMethod

Property Name	Class Defining Property	Property Description	Value or Value Location
uint16 **Characteristics**[]	CIM_DiagnosticTest	The Characteristics of the service	0 - Unkown
string **SystemCreationClassName**	CIM_Service	The CreationClassName of the scoping System.	“IBMOS400_ComputerSystem”
string **SystemName**	CIM_Service	The Name of the scoping System.	system name
string **CreationClassName**	CIM_Service	CreationClassName indicates the name of the class or the subclass that is used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	“IBM_FileMonitorService”
string **Name**	CIM_Service	The Name property uniquely identifies the Service and provides an indication of the functionality that is managed. This functionality is described in more detail in the Description property of the object.	“FileMonitorService”
boolean **Started**	CIM_Service	Started is a Boolean that indicates whether the Service has been started (TRUE), or stopped (FALSE).	TRUE

Method List:

Method Name	Class Defining Method	Description	Value or Value Location
RunDiagnostic	CIM_DiagnosticService	The RunDiagnostic method is invoked to commence execution of a diagnostic program on a specific managed element. The input parameters specify this managed element and the settings that SHALL be applied to the diagnostic and the resultant job. The method returns a reference to the ConcreteJob instance created. A job SHALL be instantiated as a means for monitoring the diagnostic as it runs and to provide useful accounting and status information once the diagnostic has completed. **Note:** Since settings may be changed without warning, the RunDiagnostic method SHOULD immediately evaluate. This information is useful for post-mortem analysis of diagnostic results.	This method is used to create and run a file monitor task

The method **`RunDiagnostic`** is used to create and run a file monitor task. The method prototype:

uint32 **RunDiagnostic**(CIM_ManagedElement REF ManagedElement, CIM_DiagnosticSetting REF DiagSetting, CIM_JobSettingData REF JobSetting, CIM_ConcreteJob REF Job);

Input Parameter:

ManagedElement: Specifies the element upon which the DiagnosticService SHOULD be run.

Note: It should be a reference of CIM_LogicalFile that will be monitored. But since the CIM_LogicalFile has not been implemented yet on IBM i, it **SHOULD** be null and input the monitored file information via DiagSetting.

DiagSetting: Specifies the desired settings that SHOULD be applied to the Diagnostic. It should be reference of IBMi_FileContentMonitorSetting for file content monitor.

Note: Clients **SHOULD** examine the appropriate capabilities and **CREATE** valid DiagnosticSetting instances to apply as input parameters.

JobSetting: Specifies the desired settings that SHALL be applied to the resulting Job.

Note: We do **NOT** support this parameter, it **SHOULD** be null.

Output Parameter:

Job: Returns a reference to the resulting Job.

Return Code:

If 0 is returned, then the task completed successfully. Any other return code indicates an error condition.

0: Success

2: Unknown

3: Timeout

4: Failed

5: Invalid Parameter

6: Busy

Note: Prior to invoking this method, clients **SHOULD** examine the appropriate capabilities and **CREATE** valid DiagnosticSetting instances to apply as input parameters.

IBM_FileMonitorServiceCapabilities

Description: Capabilities of a CIM_DiagnosticTest.

Hierarchy:CIM_DiagnosticServiceCapabilities, CIM_EnabledLogicalElementCapabilities, CIM_Capabilities, CIM_ManagedElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames

Property name	Class Defining Property	Property Description	Value or Value Location
uint16 **Supported Service Modes**[]	CIM_DiagnosticService Capabilities	The SupportedServiceModes property lists the operational modes that are supported by the Diagnostic Service. These modes may correspond to, but are not limited to, settings that may be applied to a Diagnostic Service.	0x8000 - No Service Modes
uint16 **Supported Loop Control**[]	CIM_DiagnosticService Capabilities	The SupportedLoopControl property lists the loop controls that are supported by the Diagnostic Service.	0x8000 - No Loop Control
uint16 **Supported Log Options**[]	CIM_DiagnosticService Capabilities	The SupportedLogOptions property lists the log options that are supported by the Diagnostic Service.	0x8000 - No Log Option
uint16 **Supported Log Storage**[]	CIM_DiagnosticService Capabilities	The SupportedLogStorage property lists the storage options for logging that are supported by the Diagnostic Service.	0x8000 - No Log Storage
uint16 **Supported Execution Controls**[]	CIM_DiagnosticService Capabilities	This property lists the job controls that are supported by the Diagnostic Service.	2 - Kill Job 3 - Suspend Job 4 - Terminate Job
uint16 **Supported Test Warnings**	CIM_DiagnosticService Capabilities	Lists the level of warning messages that MAY be logged.	2 - No Warnings
string **InstanceID**	CIM_Capabilities	Within the scope of the instantiating Namespace, InstanceID opaquely and uniquely identifies an instance of this class.	“FileMonitorServiceCapabilities”
String **ElementName**	CIM_Capabilities	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. **Note:** Name does not have to be unique within a namespace.	“FileMonitor Service Capabilities”

IBM_FileMonitorServiceElementCapabilities

Description: IBM_FileMonitorServiceElementCapabilities represents the association between ManagedElements and their Capabilities. The cardinality of the ManagedElement reference is Min(1), Max(1). This cardinality mandates the instantiation of the ElementCapabilities association for the referenced instance of Capabilities. ElementCapabilities describes the existence requirements and context for the referenced instance of ManagedElement. Specifically, the ManagedElement MUST exist and provides the context for the Capabilities.

Hierarchy: CIM_ElementCapabilities

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property Value (and data source)	Instance mapping rule
CIM_ManagedElement REF **ManagedElement**	The managed element.	FileMonitorService to FileMonitorServiceCapabilities
CIM_Capabilities REF **Capabilities**	The Capabilities object associated with the element.	

IBM_HostedFileMonitorService

Description: IBM_HostedFileMonitorService is an association between a Service and the System on which the functionality is located. The cardinality of this association is one-to-many. A System can host many Services. Services are weak with respect to their hosting System. Heuristic: A Service is hosted on the System where the LogicalDevices or SoftwareFeatures that implement the Service are located. The model does not represent Services hosted across multiple systems. The model is as an ApplicationSystem that acts as an aggregation point for Services that are each located on a single host.

Hierarchy: CIM_HostedService, CIM_HostedDependency, CIM_Dependency

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property name	Property value and data source	Instance mapping rule
CIM_Service REF **Dependent**	The Service hosted on the System.	Host System to FileMonitorService.
CIM_System REF **Antecedent**	The hosting System.	

IBM_OwningFileContentMonitorJob

Description: IBM_HostedFileMonitorService represents an association between a Job and the ManagedElement responsible for the creation of the Job.

Hierarchy: CIM_OwningJobElement

Intrinsic Methods: getInstance, enumerateInstances, enumerateInstanceNames, associators, associatorNames, references, referenceNames

Property Name	Property Value (and data source)	Value or Value Location
CIM_ManagedElement REF **OwningElement**	The ManagedElement responsible for the creation of the Job.	IBM_FileContentMonitorJob to IBM_FileMonitorService.
CIM_Job REF **OwnedElement**	The Job created by the ManagedElement.	

IBM_FileContentChangedIndication

Description: IBM_FileContentChangedIndication represents the event that the file content was changed ocurrs.

Hierarchy: CIM_AlertIndication, CIM_ProcessIndication, CIM_Indication

Intrinsic Methods: createSubscription, deleteSubscription

List of properties:

Property Name: String AlertingManagedElement

Class defining property: CIM_AlertIndication

Property description: The identifying information of the entity (ie, the instance) for which this Indication is generated.

Value or value location: The ObjectPath of IBM_FileMonitorService

Property Name: uint16 AlertingElementFormat

Class defining property: CIM_AlertIndication

Property description: The format of the this property is interpretable based upon the value of this property.

Value or value location: 2 - CIMObjectPath

Property Name: uint16 AlertType

Class defining property: CIM_AlertIndication

Property description: Primary classification of the Indication.

Value or value location: 8 - Security Alert

Property Name: uint16 PerceivedSeverity

Class defining property: CIM_AlertIndication

Property description: An enumerated value that describes the severity of the Alert Indication from the notifier's point of view

Value or value location: 2 - Information

Property Name: String EventID

Class defining property: CIM_AlertIndication

Property description: An instrumentation or provider specific value that describes the underlying "real-world" event represented by the Indication.

Value or value location: {IBM_FileContentMonitorJob.InstanceID}

Property Name: Datetime EventTime

Class defining property: CIM_AlertIndication

Property description: The time and date the underlying event was first detected. If specified, this property MUST be set to NULL if the creating entity is not capable of providing this information.

Value or value location: Date and time detected the file change

Property Name: String SystemCreationClassName

Class defining property: CIM_AlertIndication

Property description: The scoping System's CreationClassName for the Provider generating this Indication.

Value or value location: IBMOS400_ComputerSystem

Property Name: String SystemName

Class defining property: CIM_AlertIndication

Property description: The scoping System's Name for the Provider generating this Indication.

Value or value location: HostName

Property Name: String ProviderName

Class defining property: CIM_AlertIndication

Property description: The name of the Provider generating this Indication.

Value or value location: “UME_FileContentChangedIndicationProvider”

Property Name: String IndicationIdentifier

Class defining property: CIM_Indication

Property description: An identifier for the Indication. This property is similar to a key value in that it can be used for identification, when correlating Indications.

Value or value location: { IBMi_FileContentMonitorSetting.SettingID }

Property Name: Datetime IndicationTime

Class defining property: CIM_Indication

Property description: The time and date of creation of the Indication.

Value or value location: Indication creation time

File Content Monitor Provider Use Case

DMTF management profile, DSP 1002 Diagnostics Profile, is extended to model the file content monitor. They will provide the file content monitor management interface, which include configuration interface, lifecycle control interface and running interface.

Configure Monitor

The use cases in this section describe how the client can find and create settings for diagnostics. The IBM_/IBMi_ prefix has been omitted from the class names in the use cases for readability.

- Create file monitor settings. The client discovers the file monitor capabilities of the FileMonitorService instance. The client creates the FileContentMonitorSetting instance as necessary. However, the client should consider the file monitor capabilities during the changes.

Execute and Control Monitor

The RunDiagnostic() method is invoked to start the file monitor service. Input parameters are the ManagedElement being tested and the settings. A reference to a ConcreteJob instance is returned.

An instance of ConcreteJob is created by the file monitor provider to allow monitoring and control of the running service. By invoking the RequestStateChange method, the client may start, stop, suspend, and resume the job.

Run Monitor

The client can run a file monitor with default and unique settings as follows.

1. The client calls the RunDiagnostic( ) method, passing in references of FileContentMonitorSetting to use to execute the monitor.
2. The file monitor service creates a Job instance to represent that test running on that managed element and returns a reference to it in the return call from RunDiagnostic( ).

Suspend Monitor

The client can suspend the execution of the test by using the RequestStateChange( ) method call on the Job instance that is returned from the RunDiagnostic( ) method, as shown in the following procedure. Assume that the client starts at a known FileMonitorService instance.

1. The client follows the ElementCapabilities association from the FileMonitorService to the FileMonitorServiceCapabilities for the service.
2. The client checks the FileMonitorServiceCapabilities.SupportedExecutionControls property for the value of “Suspend Job”. If the value exists, the Job supports suspending.
3. The client finds the appropriate Job instances and calls the RequestStateChange( ) method, passing in a RequestedState value of “Suspend”.

When the transition completes successfully, the ConcreteJob that represents the monitor will set the value of the JobState property to “Suspended” and set the value of TimeOfLastStateChange to the current time.

Resume monitor

The client can resume the execution of a test by using the RequestStateChange( ) method call on the Job instance that is returned from the RunDiagnostic( ) method, as shown in the following procedure. Assume that the client starts at a known FileMonitorService instance.

1. The client follows the ElementCapabilities association from the FileMonitorService to the FileMonitorServiceCapabilities for the service.
2. The client checks the FileMonitorServiceCapabilities.SupportedExecutionControls property for the value of “Resume Job”. If the value exists, the Job supports resuming.
3. The finds the appropriate Job instances and calls the RequestStateChange( ) method, passing in a RequestedState value of “Start”.

When the transition completes successfully, the ConcreteJob that represents the test will set the value of the JobState property to “Running” and set the value of TimeOfLastStateChange to the current time.

Note: The JobState property may transition to “Starting” before the final transition to “Running”.

Abort Monitor

The client can cleanly abort the execution of a test by using the RequestStateChange( ) method call on the Job instance that is returned from the RunDiagnostic( ) method, as shown in the following procedure. Assume that the client starts at a known FileMonitorService instance.

1. The client follows the ElementCapabilities association from the FileMonitorService to the FileMonitorServiceCapabilities for the service.
2. The client checks the FileMonitorServiceCapabilities.SupportedExecutionControls property for the value of “Terminate Job”. If the value exists, the Job supports termination.
3. The client finds the appropriate Job instances and calls the RequestStateChange( ) method, passing in a RequestedState value of “Terminate”.

When the transition completes successfully, the ConcreteJob that represents the test will set the value of the EnabledState property to “Terminated” and set the value of TimeOfLastStateChange to the current time.

Note: The JobState property may transition to “Shutting Down” before the final transition to “Terminated”.

Kill Monitor

The client can immediately abort the execution of a test, with no attempt to perform a clean shutdown, by using the RequestStateChange( ) method call on the Job instance that is returned from the RunDiagnostic( ) method, as shown in the following procedure. Assume that the client starts at a known FileMonitorService instance.

1. The client follows the ElementCapabilities association from the FileMonitorService to the FileMonitorServiceCapabilities for the service.
2. The client checks the FileMonitorServiceCapabilities.SupportedExecutionControls property for the value of “Kill Job”. If the value exists, the Job supports kill.
3. The client finds the appropriate Job instances and calls the RequestStateChange( ) method, passing in a RequestedState value of “Kill”.

When the transition completes successfully, the ConcreteJob that represents the test will set the value of the EnabledState property to “Killed” and set the value of TimeOfLastStateChange to the current time.

Monitor File content modification event on the system

1. Create the subscription on the indication provider for file content modification event. For example, the query can specified “select * from IBM_FileContentChangedIndication”.
2. Start CIM listener on the destination host. And prepare to receive the CIM_AlterIndication indication from the system.

Host Hardware RAID providers

The Host Hardware Redundant Array Of Independent Disks (RAID) Controller profile is intended to represent the manageable elements of a host-based RAID controller and the storage it controls. A RAID controller can manage Small Computer System Interface (SCSI) or Advanced Technology Attachment (ATA) disks contained within a server's internal drive cage or an external drive enclosure.

In addition, a host-based RAID controller can manage physical aspects of a RAID controller card, such as battery-backed cache, audible alarms, external serial-attached SCSI (SAS), SCSI, ATA ports, or other miscellaneous sub-elements. The Host Hardware RAID Controller profile can be used to model manageability for software-based RAID included in drivers where storage volumes map to physical drives.

The Host Hardware RAID Controller profile defines the model and functions of a host where a RAID Controller resides. Several key or dependency elements in the sub profiles or packages such as the Physical Asset Profile, Block service package, DAPort subprofile, and General Initial Port profile are partially implemented.

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

Table 197. Host hardware RAID providers

Provider name	Implements CIM class	Provider type	Category
QUME_RAIDComputerSystemProvider	IBM_RAIDComputerSystem	Instance	Host hardware RAID
QUME_LogicalIdentityProvider	IBM_LogicalIdentity	Instance and Association	Host hardware RAID
QUME_RAIDPortControllerProvider	IBM_RAIDPortController	Instance	Host hardware RAID
QUME_SystemRAIDComponentProvider	IBM_SystemRAIDComponent	Instance and Association	Host hardware RAID
QUME_RAIDComputerSystemPackageProvider	IBM_RAIDComputerSystemPackage	Instance and Association	Physical asset
QUME_RAIDElementSoftwareIdentityProvider	IBM_RAIDElementSoftwareIdentity	Instance and Association	Software identity

IBM_RAIDComputerSystem

In the Host Hardware RAID Controller profile, the ComputerSystem class within this profile represents the RAID controller itself. The ComputerSystem that represents the RAID controller system acts as the principal class of the profile. Many of the other classes in the Host Hardware RAID Controller profile that together act as a host-based RAID controller are scoped to the instance of ComputerSystem that represents the controller.

Table 198. IBM_RAIDComputerSystem

Property name	Property description	Value or value location
string Caption (64)	A short textual description (one-line string) of the object.	Host Hardware RAID Controller *Name*
string Description	A textual description of the object.	Host Hardware RAID Controller *Name*
string ElementName	A user-friendly name for the object.	*Name*
string InstanceID	An optional property that can be used to opaquely and uniquely identify an instance of this class within the scope of the instantiating namespace.	*Name*

Table 198. IBM_RAIDComputerSystem (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault	An enumerated value indicating an administrator's default or startup configuration for the enabled state of an element.	2 (Enabled)
uint16 EnabledState	An integer enumeration that indicates the enabled and disabled states of an element.	2 (Enabled)
unit16 RequestedState	An integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested.	5 (No Change)
string CreationClassName(key)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_RAIDComputerSystem
string Name(key) (1024)	The key of a system instance in an enterprise environment. Name is inherited.	
string IdentifyingDescriptions[]	An array of free-form strings providing explanations and details behind the entries in the OtherIdentifyingInfo array.	Type, serial, model
string OtherIdentifyingInfo[]	Captures additional data, beyond system name information, that could be used to identify a computer system.	
string NameFormat	Defines the precedence order for assigning the system name.	Other
uint16 Dedicated[]	An enumeration that indicates the purpose to which the computer system is dedicated.	14 (block server), 29 (host-based RAID controller)

IBM_LogicalIdentity

This provider is used to associate the ComputerSystem representing the controller with PortController.

Table 199. IBM_LogicalIdentity

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF SameElement	Instance of Host Hardware RAID Controller	One-to-one association between IBM_RAIDPortController and IBM_RAIDComputerSystem
CIM_ManagedElement REF SystemElement	Instance of ComputerSystem representing the Host Hardware RAID controller	

IBM_RAIDPortController

The PortController class is the top-level and central class of the Host Hardware RAID Controller Profile. It represents an instance of a RAID controller and controls the back-end port to the storage managed by this controller.

Table 200. IBM_RAIDPortController

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

Table 200. IBM_RAIDPortController (continued)

Property name	Property description	Value or value location
uint16 ControllerType	The type or model of the port controller.	1 (other)
string OtherControllerType	string value for controller types that is not captured by ControllerType enumeration.	RAID

IBM_SystemRAIDComponent

This provider associates one ComputerSystem representing the hosting system and ComputerSystems representing the RAID controllers.

Table 201. IBM_SystemRAIDComponent

Property name	Property value and data source	Instance mapping rule
CIM_System REF GroupComponent	Instance of hosting system.	One-to-*n* association between IBMOS400_ComputerSystem and IBM_RAIDComputerSystem
CIM_ManagedSystemElement REF PartComponent	Instance of Host Hardware RAID controller.	

Physical Asset package

The physical representation of the controller is mandatory and realized by implementing the Physical Asset Profile. The Physical Asset Profile defines the set of classes and subclasses for describing the physical assets of a managed component. Most host-based RAID controllers can be described as a physical card or chip on a motherboard. The PhysicalPackage or PhysicalComponent is associated (using Realizes) to the PortController and to the ComputerSystem representing the controller (using ComputerSystemPackage).

IBM_RAIDComputerSystemPackage

This provider associates controller ComputerSystem and PhysicalPackage from the Physical Asset profile.

Table 202. IBM_RAIDComputerSystemPackage

Property name	Property value and data source	Instance mapping rule
CIM_PhysicalPackage REF Antecedent	Instance of Host Hardware RAID Controller physical card	One-to-one association between IBM_Card and IBM_RAIDComputerSystem
CIM_ComputerSystem REF Dependent	Instance of ComputerSystem representing the Host Hardware RAID controller	

Software Identity Profile

For the Host Hardware RAID Controller profile, the SoftwareIdentity class from the Software Inventory profile is required to model various software entities for a RAID controller. The implementation uses the Software Inventory profile to model the driver software for the RAID controller running on the host operating system and the firmware internal to the controller. If the RAID controller has a separate software entity for the BIOS from the firmware, the implementation can use the Software Inventory profile to represent the BIOS.

To model the driver, firmware, and BIOS software for the controller, the implementation instantiates an instance of SoftwareIdentity class associated to the top level ComputerSystem that represents the RAID controller, using the ElementSoftwareIdentity association. The SoftwareIdentity instances are differentiated by including the values Driver, Firmware, or FCode/BIOS in the *Classifications* property.

IBM_RAIDElementSoftwareIdentity

This provider associates SoftwareIdentity with controller ComputerSystem.

Table 203. IBM_RAIDElementSoftwareIdentity

Property name	Property value and data source	Instance mapping rule
CIM_SoftwareIdentity REF Antecedent	Instance of driver or firmware	*N*-to-one association between IBM_SoftwareIdentity and IBM_RAIDComputerSystem
CIM_ManagedElement REF Dependent	Instance of controller ComputerSystem	

Block services package providers

Many devices and applications provide their storage capacity to external devices and applications through block-based I/O. This subprofile defines a standard expression of existing storage capacity, the assignment of capacity to storage pools, and the allocation of capacity to be used by external devices or applications.

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

Table 204. Block services package providers

Provider name	Implements CIM class	Provider type	Category
QUME_AllocatedVolumeFromStoragePoolProvider	IBM_AllocatedVolumeFromStoragePool	Instance and Association	Block Service
QUME_AllocatedPoolFromStoragePoolProvider	IBM_AllocatedPoolFromStoragePool	Instance and Association	Block Service
QUME_PrimordialStoragePoolProvider	IBM_PrimordialStoragePool	Instance	Block Service
QUME_ConcreteStoragePoolProvider	IBM_ConcreteStoragePool	Instance	Block Service
QUME_StorageVolumeProvider	IBM_StorageVolume	Instance	Block Service
QUME_RAIDSystemVolumeDeviceProvider	IBM_RAIDSystemVolumeDevice	Instance and Association	Block Service
QUME_ConcreteComponentProvider	IBM_ConcreteComponentr	Instance and Association	Block Service
QUME_StorageCapabilitiesProvider	IBM_StorageCapabilities	Instance	Block Service
QUME_StoragePoolElementCapabilitiesProvider	IBM_StoragePoolElementCapabilities	Instance and Association	Block Service
QUME_MediaPresentProvider	IBM_MediaPresent	Instance and Association	Block Service

IBM_AllocatedVolumeFromStoragePool

This provider associates Volume or LogicalDisk from pool.

Table 205. IBM_ AllocatedVolumeFromStoragePool

Property name	Property value and data source	Instance mapping rule
CIM_StoragePool REF Antecedent	Instance of storage pool	One-to-*n* association between IBM_ConcreteStoragePool and IBM_StorageVolume
CIM_LogicalElement REF Dependent	Instance of storage volume	

IBM_AllocatedPoolFromStoragePool

This provider is used to associate the IBM_ConcreteStoragePool from IBM_PrimordialStoragePool .

Table 206. IBM_AllocatedPoolFromStoragePool

Property name	Property value and data source	Instance mapping rule
CIM_StoragePool REF Antecedent	Instance of primordial storage pool	One-to-*n* association between IBM_PrimordialStoragePool and IBM_ConcreteStoragePool.
CIM_LogicalElement REF Dependent	Instance of concrete storage pool	

IBM_PrimordialStoragePool

As required by block services package, an implementation instantiates at least one primordial storage pool that represents the physical disk storage attached to the controller. The primordial storage pool is created by the provider and cannot be deleted or modified. It cannot be used to allocate any storage element other than concrete storage pools.

This provider returns an instance of the pools of all physical disk units available on the system when a numbered list of instances is asked for, or it looks up the resource based on the Logical Resource Name provided as the key under the *InstanceID* property.

Note: To enable this class for IBM i 7.1, you need to install PTF MF50726.

Table 207. IBM_PrimordialStoragePool

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Primordial Storage Pool
string Description	Textual description of the object.	Primordial Storage Pool
string ElementName	A user-friendly name for the object.	ParitySet-*InstanceID* or unprotected pool
string Name (1024)	Defines the label by which the object is known.	ParitySet-*InstanceID* or unprotected pool
string InstanceID (key)	Within the scope of the instantiating namespace, opaquely and uniquely identifies an instance of this class.	*PoolID*
string PoolID	A unique name in the context of the system that identifies this pool.	
boolean Primordial	If true, indicates that this StoragePool is the base from which storage capacity is drawn and returned in the activity of storage management.	TRUE
uint64 RemainingManagedSpace	The remaining usable capacity after the allocation of StorageVolumes, LogicalDisks, or child storage pools.	

Table 207. IBM_PrimordialStoragePool (continued)

Property name	**Property description**	**Value or value location**
uint64 TotalManagedSpace	The total amount of capacity usable for the allocation of StorageVolumes, LogicalDisks, or child storage pools.	

IBM_ConcreteStoragePool

A concrete StoragePool is allocated from another StoragePool. It can be used for allocating StorageVolumes and LogicalDisks as well as other concrete StoragePools.

This provider returns instances of allocated storage pools available on the system when a numbered list of instances is asked for, or looks up the resource based on the Logical Resource Name provided as the key under the *InstanceID* property.

Note: To enable this class for IBM i 7.1, you need to install PTF MF50726.

Table 208. IBM_ConcreteStoragePool

Property name	**Property description**	**Value or value location**
string Caption (64)	Short textual description of the object.	Storage pool under parity or storage pool without parity
string Description	Textual description of the object.	Storage pool under parity or storage pool without parity
string ElementName	A user-friendly name for the object.	ParitySet-*InstanceID* or unprotected pool
string Name (1024)	Defines the label by which the object is known.	ParitySet-*InstanceID* or unprotected pool
string InstanceID (key)	Within the scope of the instantiating Namespace, opaquely and uniquely identifies an instance of this class.	*PoolID*
string PoolID	A unique name in the context of the system that identifies this pool.	
boolean Primordial	If true, indicates that this StoragePool is the base from which storage capacity is drawn and returned in the activity of storage management.	TRUE
uint64 RemainingManagedSpace	The remaining usable capacity after the allocation of StorageVolumes, LogicalDisks, or child storage pools.	
uint64 TotalManagedSpace	The total amount of capacity usable for the allocation of StorageVolumes, LogicalDisks, or child storage pools.	

IBM_StorageVolume

This provider returns the system storage volume. For class properties, refer to "IBM_StorageExtent" on page 228. The *CreationClassName* of this provider is IBM_StorageVolume.

IBM_RAIDSystemVolumeDevice

This provider associates System to StorageVolume or LogicalDisk.

Table 209. IBM_RAIDSystemVolumeDevice

Property name	Property value and data source	Instance mapping rule
CIM_System REF GroupComponent	Instance of system that represents host hardware controller	One-to-*n* association between IBM_RAIDComputerSystem and IBM_StorageVolume
CIM_LogicalDevice REF PartComponent	Instance of storage volume	

IBM_ConcreteComponent

This provider associates storage pool with storage extent.

Table 210. IBM_ConcreteComponent

Property name	Property value and data source	Instance mapping rule
CIM_ManagedElement REF GroupComponent	Instance of storage pool	One-to-*n* association between IBM_PrimordialStoragePool and IBM_StorageExtent
CIM_ManagedElement REF PartComponent	Instance of storage extent	

IBM_StorageCapabilities

The instantiation of StorageCapabilities is used to model the storage capabilities of the controller. This instance allows the client to easily determine the storage capabilities of the controller. This capability is fixed and may change only when new functionality is added to the controller through a firmware change or update. This instantiation of StorageCapabilities is required by the Block Services Package and defines the range of redundancy capabilities of the primordial StoragePool.

This provider describes the capabilities supported for storage pool and storage controller.

Table 211. IBM_StorageCapabilities

Property name	Property description	Value or value location
string Caption (64)	Short textual description (one-line string) of the object.	Storage Capabilities
string Description	Textual description of the object.	Storage Capabilities
string ElementName	A user-friendly name for the object.	IBM i Storage Capabilities
string InstanceID (key)	Within the scope of the instantiating namespace, opaquely and uniquely identifies an instance of this class.	RAID 5 or RAID 6 or not RAID protected

Table 211. IBM_StorageCapabilities (continued)

Property name	Property description	Value or value location
uint16 DataRedundancyDefault	Describes the default number of complete copies of data that can be maintained.	1 or 2 or 0
uint16 ElementType	Indicates the type of element to which this StorageCapabilities applies.	5 (storage pool)
boolean NoSinglePointOfFailure	Indicates whether or not the associated element supports no single point of failure.	RAID 5 or RAID 6 equal to TRUE or FALSE
uint16 ParityLayoutDefault	Specifies whether a parity-based storage organization is using rotated or non-rotated parity.	RAID 5 or RAID 6 equal to 3 or 2

IBM_StoragePoolElementCapabilities

This provider StorageCapabilities with StoragePool.

Table 212. IBM_StoragePoolElementCapabilities

Property name	Property value and data source	Instance mapping rule
CIM_Capabilities REF Capabilities	Instance of storage capabilities	One-to-one association between IBM_StorageCapabilities and IBM_PrimordialStoragePool.
CIM_ManagedElement REF ManagedElement	Instance of concrete storage pool	

IBM_MediaPresent

This provider associates storage extent and disk drive.

Table 213. IBM_MediaPresent

Property name	Property value and data source	Instance mapping rule
CIM_MediaAccessDevice REF Antecedent	Instance of disk drive	One-to-one association between IBMi_DiskDrive and IBM_StorageExtent.
CIM_StorageExtent REF Dependent	Instance of storage extent	

DAPort subprofile providers

For the Host Hardware RAID Controller profile, the Direct Attach Port profile models the port facing the host.

The DAPort class includes a property that defines the way the controller is connected to the host. The property shall be set to a valid port type as defined by the class. These include but are not limited to PCI, PCI-E, PCI express, and embedded. The ProtocolEndpoint shall be specialized to the protocol the host is using to communicate with the controller. This is normally SCSI but may be others. The DAPort also is associated to StorageVolumes (LogicalDevice) that the controller makes visible to the host.

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

Table 214. DAPort providers

Provider name	Implements CIM class	Provider type	Category
QUME_DAPortProvider	IBM_DAPort	Instance	DA Port
QUME_DAPortDeviceSAPImplementationProvider	IBM_DAPortDeviceSAPImplementation	Instance and Association	DA Port
QUME_DAPortHostedAccessPointProvider	IBM_DAPortHostedAccessPoint	Instance and Association	DA Port
QUME_DAPortProtocolEndpointProvider	IBM_DAPortProtocolEndpoint	Instance	DA Port
QUME_RAIDSystemDAPortDeviceProvider	IBM_RAIDSystemDAPortDevice	Instance and Association	DA Port

IBM_DAPort

This provider returns the direct attached port that represents the logical aspects of the physical port and might have multiple associated protocols.

Table 215. IBM_DAPort

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Direct Attached Port *DeviceID*
string Description	Textual description of the object.	Direct Attached Port *DeviceID*
string ElementName	A user-friendly name for the object.	*DeviceID*
uint16 HealthState	Indicates the current health of the element.	
string Name (1024)	Defines the label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	Indicates the current status of the element.	
string StatusDescriptions[]	Describes the various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1 ('Other').	powered off or not connected
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)
uint16 Availability	The primary availability and status of the Device.	

Table 215. IBM_DAPort (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.	
string DeviceID (Key) (64)	An address or other identifying information to uniquely name the LogicalDevice.	
string CreationClassName (Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_DAPort
uint16 PortType	Defined to force consistent naming of the 'type' property in subclasses and to guarantee unique enum values for all instances of NetworkPort.	1
string OtherPortType	Describes the type of module, when PortType is set to 1 ('Other').	DAPort

IBM_DAPortDeviceSAPImplementation

This provider associates front-end LogicalPort and target ProtocolEndpoint.

Table 216. IBM_DAPortDeviceSAPImplementation

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Antecedent	Instance of DAPort	One-to-*n* association between IBM_DAPort and IBM_DAPortProtocolEndpoint
CIM_ServiceAccessPoint REF Dependent	Instance of DAPort protocol endpoint	

IBM_DAPortHostedAccessPoint

This provider associates controller ComputerSystem to ProtocolEndpoint.

Table 217. IBM_DAPortHostedAccessPoint

Property name	Property value and data source	Instance mapping rule
CIM_System REF Antecedent	Instance of Host hardware RAID controller computer system	One-to-*n* association between IBM_RAIDComputerSystem and IBM_DAPortProtocolEndpoint
CIM_ServiceAccessPoint REF Dependent	Instance of DAPort protocol endpoint	

IBM_DAPortProtocolEndpoint

This provider represents a protocol (command set) associated to a port.

Table 218. IBM_DAPortProtocolEndpoint

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	DAPort SCSIProtocolEndpoint *ElementName*
string ElementName	A user-friendly name for the object.	*Name*
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5(No change)
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	7(No default)
string SystemName(Key) (256)	The name of the scoping system.	
string SystemCreationClassName(Key) (256)	The CreationClassName of the scoping system.	IBMOS400_ComputerSystem
string CreationClassName(Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_DAPortProtocolEndpoint
string Description	Textual description of the object.	DAPort SCSIProtocolEndpoint *ElementName*
string NameFormat (256)	Contains the naming heuristic that is chosen to ensure that the value of the Name property is unique.	ResourceName
string Name(Key) (1024)	A string that identifies this ProtocolEndpoint with either a port or an interface on a device.	
uint16 Role	Indicates which role this ProtocolEndpoint implements.	3

IBM_RAIDSystemDAPortDevice

This provider associates controller ComputerSystem to LogicalPort.

Table 219. IBM_RAIDSystemDAPortDevice

Property name	Property value and data source	Instance mapping rule
CIM_System REF GroupComponent	Instance of system represents host hardware controller	One-to-one association between IBM_RAIDComputerSystem and DAPort.
CIM_LogicalDevice REF PartComponent	Instance of DAPort	

Generic Initiator Ports providers

The Host Hardware RAID Controller profile uses the Generic Initiator Ports profile to model the back-end ports of the controller that are connected to the storage managed by the RAID controller.

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

Table 220. Generic Initiator Ports providers

Provider name	Implements CIM class	Provider type	Category
QUME_HBADeviceSAPImplementationProvider	IBM_HBADeviceSAPImplementation	Instance and Association	Generic Initiator Port
QUME_RAIDHostedAccessPointProvider	IBM_RAIDHostedAccessPoint	Instance and Association	Generic Initiator Port
QUME_SPIPortProvider	IBM_SPIPort	Instance	Generic Initiator Port
QUME_RAIDSystemDeviceProvider	IBM_DAPortProtocolEndpoint	Instance and Association	Generic Initiator Port
QUME_RAIDControlledByProvider	IBM_RAIDControlledBy	Instance and Association	Generic Initiator Port
QUME_DiskDriveSAPImplementationProvider	IBM_DiskDriveSAPImplementation	Instance and Association	Generic Initiator Port
QUME_HBAProtocolEndpointProvider	IBM_HBAProtocolEndpoint	Instance	Generic Initiator Port
QUME_DiskDriveProtocolEndpointProvider	IBM_DiskDriveProtocolEndpoint	Instance	Generic Initiator Port

IBM_HBADeviceSAPImplementation

This provider associates RAID Controller to Generic Initiator Port.

Table 221. IBM_HBADeviceSAPImplementation

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Antecedent	Instance of SPI port	One-to-one association between IBM_SPIPort and IBM_HBAProtocolEndpoint
CIM_ServiceAccessPoint REF Dependent	Instance of RAID port protocol endpoint	

IBM_RAIDHostedAccessPoint

This provider associates RAID controller ComputerSystem to initiator RAID ProtocolEndpoint..

Table 222. IBM_RAIDHostedAccessPoint

Property name	Property value and data source	Instance mapping rule
CIM_System REF Antecedent	Instance of host hardware RAID controller computer system	One-to-*n* association between IBM_RAIDComputerSystem and IBM_HBAProtocolEndpoint
CIM_ServiceAccessPoint REF Dependent	Instance of RAID initiator protocol endpoint	

IBM_SPIPort

This provider returns RAID port representing the logical aspects of the IOA and might have multiple associated protocols.

Table 223. IBM_SPIPort

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Generic Initiator Port *DeviceID*

Table 223. IBM_SPIPort (continued)

Property name	Property description	Value or value location
string Description	Textual description of the object.	Generic Initiator Port *DeviceID*
string ElementName	A user-friendly name for the object.	*DeviceID*
uint16 HealthState	Indicates the current health of the element.	
string Name (1024)	Defines the label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	Indicates the current status of the element.	
string StatusDescriptions[]	Describes the various OperationalStatus array values.	
uint16 EnabledDefault = 2	An enumerated value indicating 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.	
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1 (Other).	powered off or not connected
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
string DeviceID (Key) (64)	An address or other identifying information to uniquely name the LogicalDevice.	
string CreationClassName (Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_SPIPort
uint16 PortType	Defined to force consistent naming of the type property in subclasses and to guarantee unique enumeration values for all instances of NetworkPort.	1
string OtherPortType	Describes the type of module, when Generic Initiator PortPortType is set to 1 (Other).	DAPort

IBM_RAIDSystemDevice

This provider associates RAID controller ComputerSystem to SPI RAID Port..

Table 224. IBM_RAIDSystemDevice

Property name	Property value and data source	Instance mapping rule
CIM_System REF GroupComponent	Instance of system represents host hardware RAID controller	One-to-*n* association between IBM_RAIDComputerSystem and IBM_SPIPort
CIM_LogicalDevice REF PartComponent	Instance of SPI port	

IBM_RAIDControlledBy

This provider associates PortController to LogicalPorts.

Table 225. IBM_RAIDControlledBy

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Dependent	Returns a reference to the CIM_LogicalDevice representing a logical port.	*N*-to-one association between device (IBM_SPIPort) and controller (IBM_RAIDPortController)
CIM_Controller REF Antecedent	Returns a reference to the CIM_Controller representing a controller.	

IBM_DiskDriveSAPImplementation

This provider associates disk drive and target ProtocolEndpoint.

Table 226. IBM_DiskDriveSAPImplementation

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Antecedent	Instance of disk drive	One-to-one association between IBMi_DiskDrive and IBM_DiskDriveProtocolEndpoint
CIM_ServiceAccessPoint REF Dependent	Instance of disk drive protocol endpoint	

IBM_HBAProtocolEndpoint

This provider represents a protocol (command set) associated to a port at the initiator HBA port part.

Table 227. IBM_HBAProtocolEndpoint

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Protocol Endpoint on HBA *ElementName*
string ElementName	A user-friendly name for the object.	*Name*
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)

Table 227. IBM_HBAProtocolEndpoint (continued)

Property name	Property description	Value or value location
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the enabled state of an element.	7 (No default)
string SystemName (Key)(256)	The name of the scoping system.	
string SystemCreationClassName(Key) (256)	The CreationClassName of the scoping system	IBMOS400_ComputerSystem
string CreationClassName(Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_HBAProtocolEndpoint
string Description	Textual description of the object.	Protocol Endpoint on HBA *ElementName*
string NameFormat (256)	Contains the naming heuristic that is chosen to ensure that the value of the name property is unique.	ResourceName
string Name(Key) (1024)	A string that identifies this ProtocolEndpoint with either a port or an interface on a device.	
uint16 Role	Indicates which role this ProtocolEndpoint implements.	2

Storage HBA profile providers

The Storage HBA profile represents the manageable elements of a host bus adapter (HBA) and optionally, the storage connected to it. An HBA can be connected to disks contained within a server's internal drive cage or an external drive enclosure or array. The profile does not include enclosure management of storage devices connected to the HBA.

The PortController class is the central class of the Storage HBA Profile. It represents an instance of an HBA. The PortController is associated to one or more instances of LogicalPort (defined in the initiator port profiles) using the ControlledBy association. The PortController is associated to the ComputerSystem (from a referencing profile) using the SystemDevice association. The PortController is also associated to Product using the ProductElementComponent association; properties of Product provide information about the HBA's manufacturer and model.

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

Table 228. Storage HBA profile providers

Provider name	Implements CIM class	Provider type	Category
QUME_StorageHBAControlledByProvider	IBM_StorageHBAControlledBy	Instance and Association	Storage HBA
QUME_StorageHBAProvider	IBM_StorageHBA	Instance	Storage HBA
QUME_ProductElementComponentProvider	IBM_ProductElementComponent	Instance and Association	Storage HBA

Table 228. Storage HBA profile providers (continued)

Provider name	Implements CIM class	Provider type	Category
QUME_StorageHBARegisteredProfileProvider	IBM_StorageHBARegisteredProfile	Instance	Storage HBA
QUME_StorageHBAElementConformsToProfileProvider	IBM_StorageHBAElementConformsToProfile	Instance and Association	Storage HBA

IBM_StorageHBAControlledBy

This provider associates Storage controller to HBA Ports.

Table 229. IBM_StorageHBAControlledBy

Property name	Property value and data source	Instance mapping rule
CIM_LogicalDevice REF Antecedent	Returns a reference to the CIM_LogicalDevice representing HBA port.	*N*-to-one association between Logical Port (SPIPort) and controller (IBM_StorageHBA).
CIM_Controller REF Antecedent	Returns a reference to the CIM_Controller representing a storage controller.	

IBM_StorageHBA

This provider serves as Storage Host Bus Adapter including both RAID adapter and non-RAID adapter.

Table 230. IBM_StorageHBA

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Storage HBA *DeviceID*
string Description	Textual description of the object.	Storage HBA *DeviceID*
string ElementName	A user-friendly name for the object.	*Name*
string Name (1024)	Defines the label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	Indicates the current status of the element.	
uint16 HealthState	Indicates the current health of the element.	
uint16 EnabledDefault = 2	An enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element.	7
uint16 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5
uint16 EnabledState = 5	An integer enumeration that indicates the enabled and disabled states of an element.	

Table 230. IBM_StorageHBA (continued)

Property name	Property description	Value or value location
string OtherEnabledState	A string that describes the enabled or disabled state of the element when the EnabledState property is set to 1 ('Other').	powered off or not connected
string SystemName (Key) (256)	The scoping system's name.	
string DeviceID (Key) (64)	An address or other identifying information to uniquely name the LogicalDevice.	
string CreationClassName (Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBM_StorageHBA
uint16 Availability	The primary availability and status of the Device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
uint16 ControllerType	The type or model of the port controller.	1 (other)
string OtherControllerType	The string value for controller types that is not captured by the ControllerType enumeration.	Storage HBA

IBM_ProductElementComponent

This provider associates HBA controller to Physical Product.

Table 231. IBM_ProductElementComponent

Property name	Property value and data source	Instance mapping rule
CIM_Product REF GroupComponent	Instance of Product	One-to-one association between IBM_Product and IBM_StorageHBA
CIM_ManagedElement REF PartComponent	Instance of storage HBA	

IBM_StorageHBARegisteredProfile

This provider returns instances of registered profile for Storage HBA Profile.

Note: IBM_StorageHBARegisteredProfile is registered into the `root/PG_InterOp` namespace.

Table 232. IBM_StorageHBARegisteredProfile

Property name	Property description	Value or value location
string Caption (64)	Short textual description (one-line string) of the object.	IBM i Storage HBA Registered Profile
string Description	Textual description of the object.	IBM i Storage HBA Registered Profile
string ElementName	A user-friendly name for the object.	Storage HBA Registered Profile

Table 232. IBM_StorageHBARegisteredProfile (continued)		
Property name	**Property description**	**Value or value location**
string InstanceID (key)	Within the scope of the instantiating namespace, opaquely and uniquely identifies an instance of this class.	SNIA: Storage HBA
string RegisteredName	The name of this registered profile.	SNIA: Storage HBA Registered Profile
uint16 RegisteredOrganization	The organization that defines this profile.	11
string RegisteredVersion	The version of this profile.	1.4.0
uint16 AdvertiseTypes[]	This property signifies the advertisement for the profile information.	3 (SLP)

IBM_StorageHBAElementConformsToProfile

This provider returns the association between Storage HBA Profile and HBA Port Controller. Only the association from IBM_StorageHBAElementConformsToProfile to IBM_StorageHBA is supported..

Note: IBM_StorageHBAElementConformsToProfile is registered into the `root/PG_InterOp` namespace.

Table 233. IBM_StorageHBAElementConformsToProfile		
Property name	**Property value and data source**	**Instance mapping rule**
CIM_RegisteredProfile REF ConformantStandard	Instance of Storage HBA Profilet	One-to-*n* association between IBM_StorageHBARegisteredProfile and IBM_StorageHBA
CIM_ManagedElement REF ManagedElement	Returns instance of HBA Controller	

IOA Cache Battery providers

CIM providers are created to populate battery information and fit into the CIM data model.

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

Table 234. IOA Cache Battery providers			
Provider name	**Implements CIM class**	**Provider type**	**Category**
QUME_IOACacheBatteryProvider	IBMi_IOACacheBattery	Instance	Battery
QUME_CacheBatteryEventProvider	IBMi_CacheBatteryEvent	Indication	Battery
QUME_IOACacheBatteryMetricDefinitionProvider	IBMi_IOACacheBatteryMetricDefinition	Instance	Battery
QUME_IOACacheBatteryMetricValueProvider	IBMi_IOACacheBatteryMetricValue	Instance	Battery
QUME_IOACacheBatteryMetricInstanceProvider	IBMi_IOACacheBatteryMetricInstance	Association	Battery
QUME_IOACacheBatteryMetricDefForMEProvider	IBMi_IOACacheBatteryMetricDefForME	Association	Battery
QUME_IOACacheBatteryMetricForMEProvider	IBMi_IOACacheBatteryMetricForME	Association	Battery

IBMi_IOACacheBattery

This provider returns cache battery information on IOA.

Note: This provider requires a dependency PTF which are shipped as PTF SI41679 and MF54555 for IBM i 6.1; MF54553 for IBM I 6.1.1; MF54371 and SI41680 for IBM i 7.1.

Table 235. IBMi_IOACacheBattery

Property name	Property description	Value or value location
string Caption (64)	Short textual description of the object.	Cache Battery for IOA *DeviceID*
string Description	Textual description of the object.	Cache Battery for IOA *DeviceID*
string ElementName	A user-friendly name for the object.	*DeviceID*
uint16 HealthState	Indicates the current health of the element.	
string Name (1024)	Defines the label by which the object is known.	*DeviceID*
uint16 OperationalStatus[]	Indicates the current status of the element.	
uint16 EnabledDefault = 2	An enumerated value indicating 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 RequestedState = 12	An integer enumeration that indicates the last requested or desired state for the element.	5 (No change)
uint16 Availability	The primary availability and status of the device.	
string SystemCreationClassName (Key) (256)	The scoping system's CreationClassName.	IBMOS400_ComputerSystem
string SystemName (Key) (256)	The scoping system's name.	
string IdentifyingDescriptions [ ]	An array of free-form strings providing explanations and details behind the entries in the OtherIdentifyingInfo array.	Type, Serial, Model
string OtherIdentifyingInfo [ ] (256)	Captures additional data, beyond DeviceID information, that could be used to identify a LogicalDevice.	
string DeviceID (Key) (64)	An address or other identifying information to uniquely name the LogicalDevice.	
string CreationClassName (Key) (256)	Indicates the name of the class or the subclass used in the creation of an instance.	IBMi_IOACacheBattery
uint16 Chemistry	An enumeration that describes the chemistry of the battery.	

Table 235. IBMi_IOACacheBattery (continued)

Property name	Property description	Value or value location
string PhysicalLocation	The physical location of the I/O adapter.	
string FrameID	The frame identifier of the position of the I/O adapter.	
string CardPosition	The card position of the I/O adapter.	
uint8 Maintainable	An indicator showing whether the IOA cache battery can be concurrently replaced on the IOA.	
Boolean CacheWritten	An indicator showing if the data in the IOA cache has been successfully written onto the disk media	
uint16 DaysToWarning	The number of days until the battery sends a warning about its condition.	
uint16 DaysToError	The number of days until the battery goes into an error status, thus stopping the IOA caching function.	
uint16 PowerOnDays	The number of days that the IOA cache battery has been powered on.	

IBMi_CacheBatteryEvent

This is an event provider to monitor when a battery goes into error or warning state.

Table 236. IBMi_CacheBatteryEvent

Property name	Property description	Value or value location
string IndicationIdentifier	An identifier for the indication. This property is similar to a key value in that it can be used for identification, when correlating Indications.	IBMi_CacheBatteryEvent
datetime IndicationTime	The time and date of creation of the Indication.	Current time of creating the indication
uint16 AlertingElementFormat	The format of the AlertingManagedElement property is interpretable based upon the value of this property.	2 (Other)
string AlertingManagedElement	The identifying information of the entity for which this indication is generated.	
uint16 AlertType	Primary classification of the indication.	5 (Device Alert)

Table 236. IBMi_CacheBatteryEvent (continued)

Property name	Property description	Value or value location
string Description	A short description of the indication.	IBM i IOA Cache Battery Indication
string EventID	An instrumentation or provider specific value that describes the underlying "real world" event represented by the indication.	Object path of IBMi_IOACacheBattery
datetime EventTime	The time and date the underlying event was first detected.	Current time of creating the indication
string OtherAlertingElementFormat	A string defining other values for AlertingElementFormat.	IOA name
uint16 PerceivedSeverity	An enumerated value that describes the severity of the indication from the notifier's point of view.	6 (Critical) or 2 (Information)
uint16 ProbableCause	An enumerated value that describes the probable cause of the situation which resulted in the AlertIndication.	See Note.
string ProbableCauseDescription	Provides additional information related to the ProbableCause.	The days before IOA cache battery goes into error is less than *ProbableCause* days. or The days before IOA cache battery goes into error is more than *ProbableCause* days.
string ProviderName	The name of the provider generating this indication.	QUME_CacheBatteryEventProvider
string SystemCreationClassName	The scoping system's CreationClassName for the provider generating this Indication.	IBMOS400_ComputerSystem
string SystemName	The scoping system's name for the provider generating this indication.	Host name
uint16 ErrorState	Indicates the current health of the element.	
string PhysicalLocation	The physical location of the I/O adapter.	
uint16 DaysToWarning	The number of days until the battery sends a warning about its condition.	
uint16 DaysToError	The number of days until the battery goes into an error status, thus stopping the IOA caching function.	

Table 236. IBMi_CacheBatteryEvent (continued)

Property name	Property description	Value or value location
uint16 AdjustedPowerOnDays	The number of days that the IOA cache battery has been powered on, which the IOA has adjusted based on an internal IOA algorithm.	
uint16 PowerOnDays	The number of days that the IOA cache battery has been powered on.	
uint8 Maintainable	An indicator showing whether the IOA cache battery can be concurrently replaced on the IOA.	

Note: The default value of ProbableCause is 90 days. This value can be changed to user defined ProbableCause value by subscription SQL. For example, if ProbableCause is equal to 55, it covers the following two situations: (1) If the days before this battery goes into error is less than 55, then an event is sent; (2) If the event has been sent and days before this battery goes into error recover to more than 55, then a reset event is sent.

IBMi_IOACacheBatteryMetricDefinition

An IBMi_IOACacheBatteryMetricDefinition instance represents the definition aspects of a metric. The purpose of IBMi_IOACacheBatteryMetricDefinition is to provide a convenient mechanism for introducing a metrics definition for the IOA cache battery's days to error and days to warning.

Table 237. IBMi_IOACacheBatteryMetricDefinition

Property name	Property description	Value or value location
string Id(key)	A string that uniquely identifies the metric definition. The use of OSF UUID/GUIDs is recommended.	MDCB01 or MDCB02
string Name	The name of the metric. This name does not have to be unique, but should be descriptive and may contain blanks.	DaysToError or DaysToWarning
uint16 DataType	The data type of the metric.	11 (uint16)
uint16 Calculable	An enumerated value that describes the characteristics of the metric, for purposes of performing calculations.	3 (Non-summable)
string Units	Identifies the specific units of a value, like bytes or packets.	Days
boolean IsContinuous	Indicates whether or not the metric value is continuous or scalar. Performance metrics are an example of a linear metric.	TRUE

Table 237. IBMi_IOACacheBatteryMetricDefinition (continued)		
Property name	**Property description**	**Value or value location**
uint16 ChangeType	Indicates how the metric value changes, in the form of typical combinations of finer grain attributes such as direction change, minimum and maximum values, and wrapping semantics	4 (Gauge)
uint16 TimeScope	Indicates the time scope to which the metric value applies.	2 (point)
uint16 GatheringType	Indicates how the metric values are gathered by the underlying instrumentation. This allows the client application to choose the right metric for the purpose.	4 (OnRequest)
string ElementName	The user friendly name for this instance of Capabilities. In addition, the user friendly name can be used as a index property for a search of query. **Note:** Name does not have to be unique within a namespace.	Days to error or Days to warning
string Caption	A short textual description (one-line string) of the object.	Days to error or Days to warning
string Description	A textual description of the object.	The number of days until the battery goes into an error status, thus stopping the IOA caching function. Or The number of days until the battery sends a warning about its condition.
string InstanceID	An optional property that can be used to opaquely and uniquely identify an instance of this class within the scope of the instantiating namespace	MDCB01 or MDCB02

IBMi_IOACacheBatteryMetricValue

Each instance of IBMi_IOACacheBatteryMetricValue represents a metric value.

Table 238. IBMi_IOACacheBatteryMetricValue

Property name	Property description	Value or value location
string InstanceID(key)	Within the scope of the instantiating namespace, opaquely and uniquely identifies an instance of this class.	OA Type, IOA Model, IOA Serial Number plus *MetricDefinitionId*
string MetricDefinitionId	The key of the BaseMetricDefinition instance for this CIM_BaseMetricValue instance value.	MDCB01 or MDCB02

Table 238. IBMi_IOACacheBatteryMetricValue (continued)

Property name	Property description	Value or value location
string MeasuredElementName	A descriptive name for the element to which the metric value belongs (that is, the measured element).	
datetime TimeStamp	Identifies the time when the value of a metric instance is computed. Note that this is different from the time when the instance is created.	The date and time of the data retrieval.
string MetricValue	The value of the metric represented as a string. Its original data type is specified in CIM_BaseMetricDefinition.	
boolean Volatile	If true, indicates that the value for the next point in time can use the same object and just change its properties (such as the value or timestamp). If false, the existing objects remain unchanged and a new object is created for the new point in time.	TRUE

IBMi_IOACacheBatteryMetricInstance

The IBMi_IOACachebatteryMetricInstance class helps find the values of the definition.

Table 239. IBMi_IOACacheBatteryMetricInstance

Property name	Property value and data source	Instance mapping rule
IBMi_IOACacheBatteryBaseMetricValue REF Dependent	A CIM_BaseMetricValue instance holding the value.	One-to-*n* association between IBMi_IOACacheBatteryMetricDefinition and IBMi_IOACacheBatteryMetricValue
IBMi_IOACacheBatteryMetricDefinition REF Antecedent	The CIM_BaseMetricDefinition for this particular CIM_BaseMetricValue.	

IBMi_IOACacheBatteryMetricDefForME

The IBMi_IOACacheBatteryMetricDefForME class helps find the Managed element of the definition.

Table 240. IBMi_IOACacheBatteryMetricDefForME

Property name	Property value and data source	Instance mapping rule
IBMi_IOACacheBatteryMetricDefinition REF Dependent	A CIM_BaseMetricDefinition instance holding the metric definition	One-to-*n* association between IBMi_IOACacheBattery and IBMi_IOACacheBatteryMetricDefinition
IBMi_IOACacheBattery REF Antecedent	The managed element for this definition.	

IBMi_IOACacheBatteryMetricForME

The IBMi_IOACacheBatteryMetricForME class helps find the Managed element of the definition.

Table 241. IBMi_IOACacheBatteryMetricForME

Property name	Property value and data source	Instance mapping rule
IBMi_IOACacheBatteryMetricValue REF Dependent	A CIM_BaseMetricValue instance holding the value.	One-to-*n* association between IBMi_IOACacheBattery and IBMi_IOACacheBatteryMetricValue
IBMi_IOACacheBattery REF Antecedent	The managed element for this value.	

Message queue provider

For IBM i, all messages on the system are sent to a message queue. A message queue provider follows the DSP 1054 Indication Profile, monitoring messages in the message queue. This provider allows IBM Systems Director and any clients that follow the CIM standard to subscribe to and receive message queue indications using the CIM indication provider.

This topic introduces properties, property descriptions, and values of this provider. The following table lists the implemented CIM class, provider types, and categories for the providers.

Table 242. Message queue provider

Provider name	Implements CIM class	Provider type	Category
QUME_MessageQueueProvider	IBM_MessageQueueIndication	Indication	Event

IBM_MessageQueueIndication

This provider monitors message queues to which the user has subscribed, and sends out indications to its client.

The system has different types of message queues:

- workstation message queue
- user profile message queue
- job message queue
- system operation message queue
- history log message queue

In addition to these message queues, users can create their own user message queues for sending messages to system users and between application programs.

Note: This provider does not monitor the job message queue. It ignores the subscription of such kinds of events.

Table 243. IBM_MessageQueueIndication

Property name	Property description	Value or value location
string IndicationIdentifier	An identifier for the indication. This property is similar to a key value in that it can be used for identification, when correlating Indications.	
datetime IndicationTime	The time and date of creation of the indication.	

Table 243. IBM_MessageQueueIndication (continued)

Property name	Property description	Value or value location
uint16 PerceivedServerity	An enumerated value that describes the severity of the indication from the notifier's point of view.	
string Description	A short description of the indication.	IBM i message queue indication plus {IndicationIdentifier}
string AlertingManagedElement	The identifying information of the entity for which this indication is generated.	Message queue library + / + Message queue name
uint16 AlertingElementFormat	The format of the AlertingManagedElement property is interpretable based upon the value of this property.	2 (Other)
string OtherAlertingElementFormat	A string defining “Other” values for AlertingElementFormat.	Library/Name
uint16 AlertType	Primary classification of the indication.	2 (Communications Alert)
uint16 ProbableCause	An enumerated value that describes the probable cause of the situation which resulted in the AlertIndication.	1 (Other)
string ProbableCauseDescription	Provides additional information related to the ProbableCause.	
string[] RecommendedActions	Free-form description of the recommended actions to take to resolve the cause of the notification.	
string EventID	An instrumentation- or provider-specific value that describes the underlying "real-world" event represented by the indication.	
datetime EventTime	The time and date that the underlying event was first detected.	
string SystemCreationClassName	The scoping system's CreationClassName for the provider generating this indication.	IBMOS400_ComputerSystem
string SystemName	The scoping system's name for the provider generating this Indication	Host name
string ProviderName	The name of the provider generating this indication.	The name of the provider generating this indication.
string OwningEntity	A string that uniquely identifies the entity that owns the definition of the format of the message described in this instance.	IBM i

Table 243. IBM_MessageQueueIndication (continued)

Property name	Property description	Value or value location
string MessageID	A string that uniquely identifies, within the scope of the OwningEntity, the format of the message.	
string Message	The formatted message.	
string[] MessageArguments	An array containing the dynamic content of the message.	
string MessageType	The type of message.	
string MessageKey	The key of the message	
uint16 MessageSeverity	The severity of the message. Possible values are 0 through 99.	
string SenderJob	The name of the job that sent the message.	
string SenderProgram	The name of the program that sent the message.	
string AlertOption	Whether and when an SNA alert is created and sent for the message.	
string ProblemIdentification	The number the system generates to identify a problem if problem analysis can be run for the message.	
string DefaultReply	The text of the default reply to the message.	
string ReplyStatus	The reply status of the message.	
boolean IsCriticalBreakMessage	Whether the message was set by the operating system as a critical break message.	

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.

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

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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 or any equivalent agreement between us.

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.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample 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_.

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 i.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Product Number: 5770-SS1