



IBM Systems - iSeries
Systems management
Common Information Model

Version 5 Release 4





IBM Systems - iSeries
Systems management
Common Information Model

Version 5 Release 4

Note

Before using this information and the product it supports, read the information in “Notices,” on page 45.

Second Edition (February 2006)

This edition applies to version 5, release 4, modification 0 of IBM i5/OS (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

© Copyright International Business Machines Corporation 2004, 2006. All rights reserved.

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

Contents

Common Information Model	1		Troubleshoot the CIM server.	19
What's new for V5R4	2		The CIM server does not start	19
Printable PDF	2		The CIM server does not behave as expected	20
Configure Pegasus	3		Pegasus command-line utilities	20
Ensure that i5/OS has the required installation options	3		cimmoftl usage information	21
Set the required configuration parameters	3		cimconfig usage information.	23
Grant users the authorizations required to work with Pegasus	12		cimprovider usage information	25
Start the CIMOM job	12		ssltrustmgr usage information	27
Secure Pegasus	12		Pegasus developer's reference	29
Create an SSL key and certificate for Pegasus	12		Supported CIM base operating system classes	30
Configure the CIM server to verify client certificates	14		i5/OS metrics	39
Authentication	15		i5/OS support for the CIM indication provider	42
Enable Kerberos	16		Related information for CIM.	42
Authorize Pegasus	16			
Backup and recovery considerations	17		Appendix. Notices	45
Restore corrupted files.	18		Programming Interface Information	47
			Trademarks	47
			Terms and conditions	47

Common Information Model

The Common Information Model (CIM) is a standard developed by a consortium of major hardware and software vendors (including IBM®) called the Distributed Management Task Force (DMTF) as 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, and using 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, providing 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.

| CIM on IBM i5/OS™ V5R4 includes:

- Instrumentation for server resources on the system called *providers*. The providers, which are based on a subset of the standardized CIM classes, gather data on a system. CIM clients can work with this data by accessing the providers through the Common Information Model Object Manager (CIMOM).
- | • An open source implementation of the CIMOM called *Pegasus* (version 2.5) that manages
| communication between clients and providers. The CIMOM also provides several management
| functions, including security, and a set of commands that provide configuration and management
| functions to administrators.
- | • A schema (CIM schema version 2.9) that defines an information model for representing systems
| management functions.
- An implementation of the standardized formats for communication between clients and the CIMOM, called CIM in XML, V2.1 and CIM Operations over HTTP, V1.1. For more information about these standards, see the WBEM Web site.

For more information about the CIM standard, see the Introduction to CIM and the CIM Specification 2.2 on the DMTF Web site.

Each IBM operating system supports Pegasus (version 2.5). Pegasus supports most of the CIM operations defined in the CIM Operations over HTTP specification by the DMTF. For more information about what is supported in i5/OS, see to the Developer's reference provided below.

Pegasus is an open-source CIM implementation supported by each IBM operating system. For information about Pegasus support on other IBM operating systems, see CIM in the IBM Systems Software Information Center. For information about the open source Pegasus project, see the Pegasus Web site.

With support for Pegasus on systems running i5/OS V5R3 or later, users have the ability to access iSeries™ server resources through an extendible, industry-standard model.

Related information

WBEM Web site

Introduction to CIM

CIM Specification 2.2

Pegasus

What's new for V5R4

This topic highlights the changes made to this topic collection for V5R4.

For V5R4, i5/OS supports Pegasus version 2.5. With support for Pegasus version 2.5, i5/OS has security enhancements, configuration enhancements, and added CIM base operating system classes.

Configuration enhancements

The `cimconfig` command has new basic and advanced startup options. In addition, the `cimmofl` and `cimprovider` commands have been enhanced with new options.

- “Basic startup options for the `cimconfig` command” on page 4
- “Advanced startup options for the `cimconfig` command” on page 5
- “`cimconfig` usage information” on page 23
- “`cimmofl` usage information” on page 21
- “`cimprovider` usage information” on page 25

Security enhancements

Pegasus version 2.5. enhances CIM security by providing support for Secure Sockets Layer (SSL).

- “Create an SSL key and certificate for Pegasus” on page 12
- “Configure the CIM server to verify client certificates” on page 14
- “`ssltrustmgr` usage information” on page 27

Enhancements to CIM base operating system classes and i5/OS metrics

Thirteen new CIM base operating system classes are now supported. Other enhancements include new i5/OS metrics and support for CIM indications.

- “Supported CIM base operating system classes” on page 30
- “i5/OS metrics” on page 39
- “i5/OS support for the CIM indication provider” on page 42



Information enhancements

New troubleshooting information and backup and recovery information has been added for V5R4.

- “Backup and recovery considerations” on page 17
- “Restore corrupted files” on page 18
- “Troubleshoot the CIM server” on page 19

How to see what's new or changed

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

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

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

Printable PDF

Use this to view and print a PDF of this information.


To view or download the PDF version of the Common Information Model topic, select Common Information Model (about 253 KB).

Saving PDF files

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

1. Right-click the PDF in your browser (right-click the link above).
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) .

Configure Pegasus

The Pegasus CIMOM provides the ability to set a number of configuration options.

To configure CIMOM, administrators need to:

- Ensure that the IBM i5/OS system has the required installation options
- Set the required configuration parameters (especially security)
- Grant users the authorizations required to work with Pegasus
- Start the CIMOM job

Related concepts

“Pegasus command-line utilities” on page 20

The open source Pegasus implementation includes a set of command line utilities that you can use to control or change the Pegasus environment.

“cimconfig usage information” on page 23

Configure the startup options for the CIMOM with the **cimconfig** command.

Ensure that i5/OS has the required installation options

CIM requires the specific installation options in i5/OS.

The options are as follows:

- Extended Base Directory Support (5722-SS1 Option 3)
- International Components for Unicode (5722-SS1 option 39)
- Qshell Interpreter (5722-SS1 Option 30)

- | **Important:** If the server is configured to use Secure Sockets Layer (SSL), then you must install OpenSSL
| (5733-SC1 Option 1). Furthermore, OpenSSL requires that you install Portable Application
| Solution Environment (PASE) (5722-SS1 Option 33) on your system.

Set the required configuration parameters

Before starting the CIM server, you should set several configuration properties using the **cimconfig** command.

- | In particular, you need to determine how you are going to specify your client authentication. You can use
| any of the following methods:
- | • Use Secure Sockets Layer (SSL) certificate-based authentication

- Use Basic or Kerberos authentication.
- Use Basic or Kerberos for authentication while using SSL to encrypt CIM data between the client and the server.

Pegasus 2.5 supports SSL for the following connections:

- External connections over SSL-secured ports for CIM Client connections.
- For the connections with a CIM Export client.

To use SSL, ensure that the Enable HTTPS Connection (`enableHttpsConnection`) property is set to `true` (the default setting). This property allows to use HTTPS TCP/IP communication. An HTTPS connection has better security than HTTP. You can use SSL by itself or with Basic or Kerberos authentication.

Set the HTTP Authentication (`httpAuthType`) property to specify the type of authentication to use. This property can be set to two values: Basic, or Kerberos. Basic authentication does not protect passwords; therefore, it is recommended that you use basic authentication only with SSL.

Remember: Most CIM clients support Basic authentication but not Kerberos authentication. Ensure that you know which type of authentication the CIM clients in your environment support before you configure the CIM server.

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

- `shutdownTimeout`
- `logLevel`
- `traceComponents`
- `traceFilePath`
- `traceLevel`

These properties are *dynamic*—that is, when you change these properties, the changes take place immediately. You do not need to restart the CIM server to carry out the changes. You can update the current configuration properties only while the CIM server is running.

All of the properties can be changed in the planned configuration properties, whether the CIM server is running or stopped. If the planned configuration properties are changed, those changes will not take effect until the CIM server is restarted. When the CIM server is started, the planned configuration properties become the current configuration properties.

Basic startup options for the `cimconfig` command

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

The following list describes the startup options and default values for the `cimconfig` command. None of the basic startup options are dynamic.

`enableHttpConnection`

Allows user access through the `httpPort`, using HTTP TCP/IP communication. The default is `false`, which means that the CIM server will not listen at the `httpPort`. Set this property to `true` if you are certain your environment is secure.

If this property is set to `false`, then `enableHttpsConnection` should be set to `true` to allow HTTPS connections.

Default value

`false`

enableHttpsConnection

Allows user access through the httpsPort, using HTTPS TCP/IP communication. The default is true because an HTTPS connection has better security than an HTTP connection.

Default value

true

httpPort

Port to listen for HTTP requests. If set, then this must be set to a valid port number, and it overrides the port number of the wbem-http service in the TCP/IP services table. If not set, then the port from the wbem-http service is used. If neither this property nor the wbem-http service port is set, then a hard-coded default of 5988 is used.

This property only takes affect if enableHttpConnection is set to true.

Default value

5988

httpsPort

Port to listen for HTTPS requests. If set, then this must be set to a valid port number, and it overrides the port number of the wbem-https service in the TCP/IP services table. If not set, then the port from the wbem-https service is used. If neither this property nor the wbem-https service port is set, then a hardcoded default of 5989 is used.

This property only takes affect if enableHttpsConnection is set to true.

Default value

5989

httpAuthType

Type of HTTP authentication (Basic, Kerberos).

Note: You should not use Basic authentication unless the one or both of the following conditions are true:

- You are using HTTPS.
- You have a highly secure environment, where passing clear-text passwords is not an issue.

This property determines the authentication to be performed over the HTTP and HTTPS ports (but see also the sslClientVerificationMode property). This property does not determine the authentication over the wbem-exp-https port (see the enableSSLEXPExportClientVerification property).

Default value

Basic

Note: If the value of the httpAuthType is set to Basic, the user's password is sent in the clear. If this option is set to an unrecognized value the httpAuthType will automatically default to Kerberos.

kerberosServiceName

If the httpAuthType is set to Kerberos, this sets the Kerberos service name for the CIMOM service, which should match the CIMOM service name configured in the key distribution center (KDC).

Default value

cimom

Advanced startup options for the cimconfig command

You can change the advanced startup options for the CIM server with the cimconfig command.

| The following list describes the advanced startup options for the cimconfig command, their default values, and whether they can be changed dynamically.

| **Note:** The shutdownTimeout, logLevel, traceLevel, traceComponents and traceFilePath settings can be changed dynamically. The others cannot. For all the other properties, you must use the **-p** parameter to indicate your change. You must then stop and restart the CIM Server for the change to take effect.

| **Important:** These are options are intended to be used only by advanced users.

| **messageDir**

| The default directory to search for the message bundles. The default value points to the shipped message bundles.

| **Default value**

| /QIBM/ProdData/OS400/CIM/msg

| **Dynamic**

| No

| **logLevel**

| Sets the level of data logged. Set to TRACE, INFORMATION, SEVERE, FATAL. The log data is saved in the QYCMCIMOM job log.

| **Default value**

| INFORMATION

| **Dynamic**

| Yes

| **enableNormalization**

| If set to true, ensures objects delivered from providers are complete and accurate. The default is false. Do not normalize objects from trusted entities. Objects from the repository, control providers, IBM shipped providers and certain vendor providers known to reliably produce valid objects should not be normalized. Only objects from 3rd party providers added to a distribution should be normalized. The values are true or false.

| **Default value**

| false

| **Dynamic**

| No

| **excludeModulesFromNormalization**

| Disables normalization for objects from specific provider modules. If enableNormalization is set to true, all provider objects will be normalized except for those on this exclusion list.

| **Default value**

| ""

| **Dynamic**

| No

| **repositoryIsDefaultInstanceProvider**

| Enables the repository component of the CIM server to provide CIM object instances by default. *Default* means that if there is no provider to service the client request for the CIM instance, then the CIM server repository is used. This includes both creating and retrieving instances. If the value of the repositoryIsDefaultInstanceProvider option is changed to false, the i5/OS providers that implement CIM metric classes will no longer function properly. The values are true or false.

| **Default value**

| true

| **Dynamic**
| No

| **enableAuthentication**

| If set to true, performs authentication before any request is allowed into the CIM server for processing. The default is true. Setting this property to false will allow unauthenticated access to the CIM server.

| Set enableAuthentication to false only if you are certain your environment is secure and if you have a very good reason.

| The values are true or false.

| **Default value**
| true

| **Dynamic**
| No

| **sslCertificateFilePath**

| Path to the CIM server's certificate file.

| This property must be set to a valid certificate if enableHttpsConnection or enableSslExportClientVerification is set to true. Note that an expired certificate is considered valid when it is loaded by the CIM server.

| If sslKeyFilePath is not specified then the CIM server will attempt to load the private key from the certificate file.

| **Default value**
| ssl/keystore/servercert.pem

| **Dynamic**
| No

| **sslKeyFilePath**

| Path to the CIM server's private key file. This property is not required to be set if the certificate specified in sslCertificateKeyPath contains the private key.

| This file is not protected by a pass phrase and must be kept in a protected directory. The value that is specified in the default value is a protected directory.

| **Default value**
| ssl/keystore/serverkey.pem

| **Dynamic**
| No

| **sslTrustStore**

| Path to the directory or file containing the trusted certificates for CIM Operation requests. The truststore can include CA certificates.

| This property must be set if sslClientVerificationMode is set to required.

| If sslClientVerificationMode is set to optional, then this property may be set to empty. In this case no certificates are trusted.

| If this property is set to an empty directory, or an empty file, then no certificates are trusted.

| If sslClientVerificationMode is set to disabled, this property is not used.

| **Default value**
| ssl/truststore/

| **Dynamic**

| No

| **exportSSLTrustStore**

| Path to the directory or file containing the trusted certificates for CIM Export requests. The truststore can include CA certificates.

| This property must be set if enableSSEExportClientVerificationMode is set to true.

| If this property is set to an empty directory, or an empty file, then no export certificates are trusted.

| This property only takes effect if enableSSEExportClientVerification is set to true.

| **Default value**

| ssl/exporttruststore/

| **Dynamic**

| No

| **crlStore**

| Path to the directory or file containing the certificate revocation lists.

| If this property is not set, set to an empty directory, or set to an empty file, then no CRLs are loaded.

| This property only takes effect if sslClientVerificationMode is set to required or optional, or enableSSEExportClientVerification is set to true.

| **Default value**

| ssl/crlstore/

| **Dynamic**

| No

| **sslClientVerificationMode**

| Sets the mode of SSL client certificate verification.

| Set to required, optional, or disabled.

| If set to required, the CIM server always requires verification of a client certificate on the HTTPS port and rejects the request if the client certificate is not trusted. The httpAuthType property is not used.

| Optional means the CIM server will verify a client certificate if available, otherwise the CIM server will use the httpAuthType setting for client verification.

| Disabled means the CIM server will always use the httpAuthType setting for client verification.

| This property is only effective if enableHttpsConnection is set to true.

| **Default value**

| optional

| **Dynamic**

| No

| **sslTrustStoreUserName**

| Identifies the username that is to be user context for the CIM Operation request when certificate authentication is used, and a username cannot be associated with a specific certificate file. The user context is the i5/OS user profile under which the provider is invoked to perform the CIM request. This property must be set to a valid user profile on i5/OS.

| If sslClientVerificationMode is set to disabled, this property has no effect.

If sslTrustStore is set to a directory, then this property has no effect. The username associated with the certificate file in the directory is the user context for the CIM operation request. The default setting for sslTrustStore is a directory.

If sslTrustStore is set to a single file, then this property must be set to a username, otherwise the CIM server will log an error and not start. In this case, ALL certificates included in the file are assigned to the username specified by sslTrustStoreUserName. This user name becomes the user context for the CIM Operation request.

Default value

""

Dynamic

No

enableSubscriptionsForNonprivilegedUsers

Set to true or false. The default is false. False means that only a user with *IOSYSCFG and *ALLOBJ authorities will be allowed to create Indication Subscriptions.

Default value

false

Dynamic

No

enableSSEXPClientVerification

Set to true or false. If true, allows export clients to connect using HTTPS on the port specified by the service name wbem-exp-https. Only CIM Export requests are allowed on this port.

Note: If the wbem-exp-https port is not defined in the system's TCP/IP services table, then the CIM server will log an error and not start. Since wbem-exp-https is an IANA standard service, it will be in the i5/OS services table by default.

If false, then no requests are allowed on the wbem-exp-https port.

Default value

true

Dynamic

No

shutdownTimeout

Set to a number of seconds. When an ENDTCPSPVR *CIMOM command is issued, the timeout is the maximum number of seconds allowed for the CIM server to complete outstanding CIM operation requests before shutting down. If the specified timeout period expires, the CIM server will shut down, even if there are still CIM operations in progress. Minimum value is 2 seconds. Default value is 10 seconds.

Default value

10

Dynamic

Yes

traceLevel

Level of debug trace. Range is 1 to 4. A traceLevel of 1 only traces function exits, the minimum trace. A trace level of 4 is the maximum trace.

Default value

1

| **Dynamic**
| Yes

| **traceFilePath**
| Path to the trace file.

| **Default value**
| /qibm/userdata/os400/cim/cimserver.trc

| **Dynamic**
| Yes

| **traceComponents**
| Components of Pegasus to trace. The valid settings are listed in “Settings for the
| traceComponents option” on page 11.

| **Default value**
| empty

| **Dynamic**
| Yes

| **enableAssociationTraversal**
| Set to true or false. The default is true. True means association traversal is enabled. False will
| disable association traversal.

| **Default value**
| true

| **Dynamic**
| No

| **enableIndicationService**
| Set to true or false. The default is true. True means the indication service is enabled. False will
| disable the indication service.

| **Default value**
| true

| **Dynamic**
| No

| **tempLocalAuthDir**
| The directory where the Pegasus server writes temporary files that it uses during local
| authentication.

| **Default value**
| /tmp

| **Dynamic**
| No

| **Related concepts**
| “ssltrustmgr usage information” on page 27
| This command provides a command-line interface to manage X.509 certificates in a trust store or a
| Certificate Revocation List (CRL).

| **Related tasks**
| “Create an SSL key and certificate for Pegasus” on page 12
| For Pegasus to run in Secure Sockets Layer (SSL) mode, a private key and certificate are required.
| Pegasus checks for its private key and certificate during startup. If those files do not exist, Pegasus
| creates its private key and a self-signed 365-day certificate. You can also create a private key and
| certificate with this information.

Settings for the traceComponents option

The following settings are valid for the traceComponents option:

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

- XmlParser
- XmlReader
- XmlWriter

Grant users the authorizations required to work with Pegasus

There are two types of CIM operations that require user authorization.

Operations that change the local CIM schema are controlled in application administration, whereas operations that change the i5/OS system objects are controlled using object authorities in i5/OS.

Related tasks

“Secure Pegasus”

Use this topic to find out about the options that are available for ensuring that the CIM server is secure.

Start the CIMOM job

Pegasus on the iSeries server is included in i5/OS V5R3 or later, and includes the supported set of providers for i5/OS, the CIM schema, and the Pegasus CIMOM.

All Pegasus functions run under a single TCP/IP server job, QYCMCIMOM. QYCMCIMOM runs in the QSYSWRK subsystem, and is not started by default. To start the CIMOM job, perform the following steps:

1. Work with the CIMOM job in iSeries Navigator by selecting **Network** → **Servers** → **TCP/IP**
2. Select **CIMOM** You can use this window to start or stop the CIMOM, and to determine whether the CIMOM starts with TCP/IP by default.

Note: You can also start the CIMOM job from the command-line interface using the command `STRTCPSVR *CIMOM`.

You can end the CIMOM job using the command `ENDTCPSVR *CIMOM`.

Secure Pegasus

Use this topic to find out about the options that are available for ensuring that the CIM server is secure.

One of the most significant concerns for a Pegasus administrator is how to configure security. This is particularly true for i5/OS because of i5/OS platform security requirements, significant functions were added to the open source implementation. In Pegasus, there are two types of security checks, authentication and authorization.

Related concepts

Enterprise Identity Mapping (EIM) topic

“cimconfig usage information” on page 23

Configure the startup options for the CIMOM with the **cimconfig** command.

Network Authentication Service topic

Hostname resolutions considerations topic

Create an SSL key and certificate for Pegasus

- | For Pegasus to run in Secure Sockets Layer (SSL) mode, a private key and certificate are required.
- | Pegasus checks for its private key and certificate during startup. If those files do not exist, Pegasus creates its private key and a self-signed 365-day certificate. You can also create a private key and certificate with this information.

- | Before you can do this procedure, you must install OpenSSL on your system (LPO 5733-SC1).

The private key and certificate are stored in paths that are defined by the `sslKeyFilePath` and `sslCertificateFilePath` configuration properties of the `cimconfig` command. You can create your own certificate and private key in these paths. Otherwise, if either the certificate or private key does not exist in these paths, then the CIM Server will create its own certificate and private key. The CIM server creates its certificate with the following attributes for the subject name:

State or Province Name: Minnesota
Locality: Rochester
Organization Name: IBM
Organizational Unit: iSeries
Common Name: *hostname of system*
Email Address:

Note:

- The Common Name field is replaced by the hostname of this system.
- The Email Address field is left blank.
- This certificate is self-signed. The expiration date of the certificate is set to 365 days from its creation date.

After these files are created you must manage the renewal and recovery of the certificate. You need to create an SSL key and certificate whenever the certificate is not valid, expired, or its security has been compromised. You can recreate the certificate by deleting the certificate file, and restarting the CIM server. The CIM server creates a new certificate that expires in 365 days.

Note: Pegasus only supports private key files without a pass-phrase. For this reason it is important to keep the private key in a protected directory. By default, the Pegasus private key is put in a directory owned by QSYS, with PUBLIC *EXCLUDE, and no private authorities. If you change the `sslKeyFilePath` property, it is recommended that this directory be protected.

Pegasus allows the OpenSSL default for its initialization (seeding) of the pseudo random number generator (PRNG). Pegasus calls the `SSL_library_init` application programming interface (API) which calls the `i5/OS Qc3GenPrns` API (Generate Pseudorandom Numbers). Pegasus on `i5/OS` will not support seeding the PRNG from a file.

One method to create a certificate and private key for Pegasus is to use the Digital Certificate Manager (DCM) on `i5/OS`.

DCM allows you to create a Pegasus server certificate that is issued by a local Certificate Authority (CA) on the `i5/OS` system, or by an external Certificate Authority.

Note that Pegasus is not integrated with DCM. You must export all certificates that are created in DCM to Pegasus. Pegasus only supports the PEM format for certificates.

To create a private key and certificate, do the following steps:

1. Create an Application definition in DCM of type server for Pegasus. Because Pegasus is not integrated with DCM, the details of the Application definition are not important. However, the recommended Application ID is `QIBM_CIMOM`.
2. Create a certificate for the Pegasus application that is issued by a CA. Make note of the subject name that you enter for Pegasus in the certificate.
3. Export the certificate from DCM to Pegasus by doing the following steps:
 - a. In the navigation frame, select **Manage Certificates** and **Export Certificates**.
 - b. Select **Server or client** as the type of certificate.
 - c. Select the certificate that you created for Pegasus and click **Export**.
 - d. Choose **File** as the export destination.

- e. For the export file name, use the directory defined by the Pegasus **sslCertificateFilePath** property, and name the file `pegasuscert.p12`. This file will be in PKCS12 format.

Note: Make sure to remember the password that you enter here. This will be used to decrypt the exported certificate later.

4. Run the OpenSSL commands to convert the certificate from PKCS12 format to Privacy Enhanced Mail (PEM) format by doing the following steps:
 - a. At an i5/OS command line, start the PASE environment by typing `CALL QP2TERM`.
 - b. Change directory to the location of the exported certificate.
 - c. Extract the certificate from the PKCS12 file and convert to PEM format by using the following OpenSSL command: `openssl pkcs12 -in pegasuscert.p12 -out pegasuscert.pem -nokeys -clcerts` This command will prompt for the password that you entered in the **DCM Export** page. The PEM file that is created might contain more than one certificate. It might contain both the Pegasus certificate and the certificate of the CA that issued the Pegasus certificate. Because Pegasus does not support this type of PEM file, the CA certificate must be removed.
 - d. Remove the CA certificate by editing the PEM file; delete all of the lines except the ones for the Pegasus certificate. The Pegasus certificate has the Pegasus subject name that you used when you created the certificate in DCM. Keep the lines of Pegasus certificate starting with `Bag Attributes` and ending with `END CERTIFICATE`.
 - e. Extract the private key from the PKCS12 file and convert to PEM format by using the following OpenSSL command: `openssl pkcs12 -in pegasuscert.p12 -out pegasuskey.pem -nocerts -nodes` This command will prompt for the password that you entered in the **DCM Export** page. The certificate and private key are now converted to PEM format.
 - f. Make the certificate available to Pegasus by placing it in the path that is defined by the **sslCertificateFilePath** property.
 - g. Make the private key available to Pegasus by placing it in the path that is defined by the **sslKeyFilePath** property.

Related concepts

“Advanced startup options for the cimconfig command” on page 5

You can change the advanced startup options for the CIM server with the cimconfig command.

Digital Certificate Manager topic collection

“Backup and recovery considerations” on page 17

Regularly back up the Pegasus repository as part of your existing backup plan. In most cases, you can recover a damaged repository by restoring the last backup copy.

Related information

OpenSSL Web site

Configure the CIM server to verify client certificates

You can configure the CIM server to use secure sockets layer (SSL) to verify client certificate's and to check certificate revocation lists (CRLs) on the main SSL port and the export SSL port.

The CIM server uses the main SSL port for CIM operation requests, such as **GetInstance** requests and **EnumerateInstance** requests. The purpose of the export SSL port is to allow CIM export requests to use automatic certificate-based authentication on a port that does not require a user name and password. CIM export requests are used to deliver CIM Indications. Because export requests do not have an associated user name, the only way to deliver secure indications is to use SSL on the export SSL port.

The CIM server can also check client certificates against a CRL.

| **Configure client certificate verification on the main SSL port**

| To configure the CIM server to verify client certificates on the main SSL port, use the **sslClientVerificationMode** property of the **cimconfig** command. You can set this property to do one of the following tasks:

- | • Disable client certificate verification
- | • Require client certificate verification
- | • Verify the client certificate if available and use the **httpAuthType** property if the certificate is not available

| With these choices, you can authenticate clients through certificate verification, Basic authentication, or Kerberos authentication.

| You can manage the certificates in the server's truststore for the main SSL port by using the **ssltrustmgr** command. In this case the trust store name is **cim_trust**

| **Configure client certificate verification on the export SSL port**

| To configure the CIM server to verify client certificates on the export SSL port, use the **enableSSExportClientVerification** property of the **cimconfig** command. When set to true, this property causes the CIM server to require that certificates are sent by export clients. The **exportSSLTrustStore** property gives the location of the truststore. In most cases, you can use the default value of the **exportSSLTrustStore** property.

| You can manage the certificates in the server's truststore for the export SSL port by using the **ssltrustmgr** command. In this case the trust store name is **export_trust**.

| **Configure client certification against a CRL**

| To configure the CIM server to verify client certificates against a CRL, use the **crlStore** property. In most cases, the default value of the **crlStore** property can be used. The CIM server checks a CRL file or directory on the local system. It does not contact a remote CIM server for the CRL. The **crlStore** property gives the location of the CRL store. The **crlStore** applies to requests that are made on the main SSL port and the export SSL port.

| **Authentication**

| Pegasus uses an authentication process to determine which users can log into the CIMOM. Unless the **enableAuthentication** property of **cimconfig** command is set to false, authentication is performed for every connection, before users can access the CIM data.

| For Pegasus on i5/OS, users log in over HTTP or HTTPS, using either Basic or Kerberos authentication. In addition for HTTPS they have a choice of logging in using certificate-based authentication

When logging in, users are authenticated using their i5/OS profile, or using Enterprise Identity Mapping (EIM).

In the open source implementation, Pegasus maintains a separate access control list (ACL) that allows users to sign in using a CIM user profile, which does not necessarily require a corresponding profile on the system. In contrast, the i5/OS implementation of Pegasus requires each user to have a profile in i5/OS. After the user has been authenticated, a user (or the user's jobs) can have access to the providers and CIM schemas managed by the CIMOM.

Enable Kerberos

Pegasus on iSeries supports both Kerberos and Enterprise Identity Mapping (EIM). To enable Kerberos, use the `cimconfig` commands to set the `httpAuthType` configuration option to Kerberos (this is the default value).

For all IBM server platforms, the Kerberos default server name is **cimom**. For i5/OS, you can also use the service name **krbsvr400**. See the Network Authentication Service topic for more information about Kerberos on i5/OS. For information about resolving the host name for Kerberos, follow the instructions in the Hostname resolutions considerations information in the Network Authentication Service topic collection.

For example, one method for setting the CIMOM service principal would be to enter the following commands:

1. On the i5/OS system where the KDC is running, add the service principal `cimom` with the following command:

```
addprinc cimom/<host>@<realm>
```

You will be prompted for the password to the KDC.

2. On each i5/OS where the CIMOM server will need to run, add the service principal `cimom` with the following command:

```
keytab add cimom/<host>@<realm>
```

You will be prompted for the password to the keytab file.

This example makes the following assumptions:

- The password in the KDC and keytab file must match.
- The host is in the case as determined by following the instructions in the Hostname resolutions considerations.

Note:

- Refer to the Keytab command information in the Network Authentication Service topic.
- If Kerberos authentication is enabled, only CIM clients that support Kerberos authentication can connect to the CIM server.

If EIM is not enabled, the Kerberos principal will be directly used as the user identity on the system where CIMOM is running. The administrator must set up matching user identities on all their systems. For example, if a customer chooses not to configure and enable EIM, then the administrator must be aware that a Kerberos principal `john` is always mapped to `john` as the local user identity.

Authorize Pegasus

A type of security check that is required for Pegasus on i5/OS is verifying that users have access to the objects they are trying to change. This process is called *authorization*.

In Pegasus, there are two types of operations that require users to have authorization to perform them: CIM class and qualifier operations, and CIM instance operations.

CIM class and qualifier operations change the local copy of the CIM schema (for example, `DeleteClass`). Users need authorization to these operations before being able to use the operations listed in the following information with systems management data provided by CIM. These operations do not change any i5/OS system objects, but because they do change the CIM schema exposed to clients, some authorization is required to use them. For the iSeries servers, authorization to these operations is controlled by Application Administration in iSeries Navigator.

To work with authorization for CIM operations in Application Administration:

1. Start iSeries Navigator.
2. From **My connections**, right-click the system you want to change.
3. Select **Application Administration**.
4. Select **Local Settings** (if available).
5. Select **Host Applications** tab.
6. Expand **CIMOM server**.
7. Add or remove a user or groups authorization to the following operations.

Application Administration allows users to be authorized to the following operations:

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

CIM instance operations let users work with the server resources modeled by the Pegasus providers. These providers are implemented as server exit programs (*SRVPGM) in i5/OS, and users require authorization to these service programs before they can use them. All of the providers included in V5R3 ship with PUBLIC *USE authority, except for the metric provider QSYS/QYCPCSMV, which is shipped with PUBLIC *EXCLUDE authority. If any providers are added that are not shipped with PUBLIC *USE authority, administrators must explicitly grant users authorization to these objects.

Backup and recovery considerations

- | Regularly back up the Pegasus repository as part of your existing backup plan. In most cases, you can recover a damaged repository by restoring the last backup copy.
- | The Pegasus repository is located in the qibm/userdata/os400/cim directory in the integrated file system. Pegasus keeps definitions of the data about managed objects and their providers in this repository. In addition, other definitions might be added by clients and providers.
- | The definitions that are installed with the CIMOM server are as follows:
 - | • root: The root namespace exists to conform to the DMTF specifications.
 - | • root/cimv2: The standard CIM Schemas go here. Also, the schemas for the shipped providers.
 - | • root/PG_Interop: This is for provider registration. This space is reserved exclusively for providers, and all providers must register here.
 - | • root/PG_Internal: This space is reserved for use by the Pegasus CIMOM server only.
 - | • root/ibmsd: The namespace owned and used by IBM Director, shipped with the base operating system.
- | Pegasus also stores two important configuration files in the qibm/userdata/os400/cim directory:
 - | • cimserver_current.conf. This file contains the current values that are not defaulted.
 - | • cimserver_planned.conf. This file contains planned values, not yet in effect and that are not defaulted.
- | **Attention:** Do not attempt to manually edit the configuration files. You must use the **cimconfig** command.

| If the files in this directory are deleted, moved, or corrupted, you need to restore them from the backup.

| **Important:**

| If the latest backed up repository is from a V5R3 release, do not restore it to a system that has
| Pegasus V2.5. After you install Pegasus V2.5 and upgrade the repository, back up the new
| repository immediately after the CIM server has been restarted.

| The server's SSL certificate and private key files are located in the
| qibm/userdata/os400/cim/ssl/keystore directory. If you do not have a backup file for the CIM server's
| SSL certificate files, you can re-create the certificates.

| The other directories that are related to the CIM server's SSL files are as follows:

- | • The server's CIM client truststore is located in /qibm/userdata/os400/cim/ssl/truststore.
- | • The server's CIM Export client truststore is located in /qibm/userdata/os400/cim/ssl/exporttruststore.
- | • The server's certificate revocation list is located in /qibm/userdata/os400/cim/ssl/crlstore.

| If you do not have backup files for truststore certificates or certificate revocation lists, then you must
| re-create and add them to the truststore or crlstore by using the **ssltrustmgr** command from qshell.

| **Note:** Use the **ssltrustmgr** command to remove corrupted certificates from the truststore. If the
| **ssltrustmgr** command fails to remove the corrupted certificates, contact your service provider.

| **Related concepts**

| "cimconfig usage information" on page 23
| Configure the startup options for the CIMOM with the **cimconfig** command.

| **Related tasks**

| "Create an SSL key and certificate for Pegasus" on page 12
| For Pegasus to run in Secure Sockets Layer (SSL) mode, a private key and certificate are required.
| Pegasus checks for its private key and certificate during startup. If those files do not exist, Pegasus
| creates its private key and a self-signed 365-day certificate. You can also create a private key and
| certificate with this information.

| **Related information**

| Backup your server topic

| **Restore corrupted files**

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

| To recover customer files, use the information in the following list.

| **Repository classes and qualifiers (static data)**

| Undo whatever was done to create the class or qualifier. For example, uninstall a client
| application or take manual steps to undo what was done. Then, put the class or qualifier back the
| same way they were before. For example, reinstall a client application. If the problem persists,
| contact your service provider.

| **Repository instances**

| Undo whatever was done to create the instance. For example, uninstall a client application or
| take manual steps to undo what was done. Then, put the class or qualifier back the same way
| they were before. For example, reinstall a client application. If the problem still persists, contact
| your service provider.

Provider registration data (also instances)

Use the **cimprovider** command to remove the provider registration. Then, use the **cimmofl** command to recompile and reregister the data. If the problem still persists, contact your service provider.

SSL Files

CIMOM SSL certificate and private key

Delete the old certificate and private key, then do the instructions in “Create an SSL key and certificate for Pegasus” on page 12 to create a new certificate and private key

Truststore and crlstore recovery

Use the **ssltrustmgr** command to re-create the truststore and crlstore.

Troubleshoot the CIM server

Use this information to troubleshoot 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.

Related information

OpenSSL documentation

The CIM server does not start

Do the following steps if the CIM server does not start.

Note: If this is the first time the CIMOM server is being started after install of Pegasus V2.5, it takes a while for the server to start. During that time, the repository is being upgraded and the user data from the previous repository is being migrated.

Ensure that the CIM server is configured correctly.

See “Set the required configuration parameters” on page 3 to configure your CIM server.

Verify whether you need to change your Kerberos configuration or change the CIM server to use basic authentication.

If you receive the following message, there is an error with your authentication. You need to fix your authentication or use another form of authentication, such as basic authentication.

```
CPDDF82: (Message appears in CIM server, QYCMCIMOM, joblog)
PGS02000: The CIM server authentication handler for Kerberos
failed to initialize properly. The CIM server is not started.
```

Note: You should not use basic authentication unless you are using HTTPS or unless you have a highly secure environment, where passing clear text passwords is not an issue. To use the **cimconfig** command to change your CIM configuration settings, QSHELL (Option 30) must be installed.

To change the authorization to basic, type the following command at a command line.

```
qsh
cd / QIBM/UserData/OS400/CIM
cimconfig -shttpAuthType=Basic -p
```

Ensure that the correct options are installed on your system.

If you receive the CPDDF80 with reason code 07 indicating the required option 39 is not installed, install Option 39 (ICU).

You might need to install LPO 5733-SC1 (OpenSSL) if the server starts with SSL disabled and the server job log contains a message saying that OpenSSL is required. In this case the server is

configured to use SSL, but OpenSSL is not installed. You can install OpenSSL or configure the server to not use SSL. The default configuration of the server is to enable SSL on both the wbem-https and wbem-exp-https ports.

In addition, if LPO 5733-SC1 had an installation error, you will also receive the message in the server job log, and the server will start with SSL disabled. You can use the CHKPRDOPT command to check the error status of the LPO in this case.

OpenSSL requires that PASE (option 33 of the Base OS) is installed. After the server starts, if PASE is not installed, and the server has SSL enabled, then OpenSSL sends a CPFB9C1 escape message to the server job log, and the server ends.

The CIM server does not behave as expected

If you are having trouble with the CIM server, see the following instructions:

Check to see if the CIM server is running by typing WRKJOB QYCMCIMOM at a command line.

If there is not an active job, start the CIM server by typing STRTCPSVR *CIMOM at a command line.

Check to see if the CIMOM repository is corrupted.

Verify whether the repository directory and configuration files exist in the QIBM/UserData/OS400/CIM directory of the integrated file system. If any of these files are missing, restore all the repository directories and files from your backup. You must use a backup of those files from the Pegasus V2.5 backup, not from a V5R3 system that did not have the Pegasus V2.5 PTFs installed.

If a backup to restore from does not exist, follow the instructions in “Restore corrupted files” on page 18.

Verify if you are attempting to process a request when the provider is not registered or enabled.

To verify if you are attempting to process a request when the provider is not registered or enabled, do the following steps:

1. Type **cimprovider -l -s** to list the name and status of the registered provider modules.
2. Type **cimprovider -l-m *module name*** to see the individual providers in that module.

Check the job log file.

To check the job log file, do the following steps:

1. Type WRKACTJOB at a command line.
2. Look in the QSYSWRK subsystem to find the QYCMCIMOM job.
3. Select 5 (work with), then 10 (Display job log, if active, on job queue, or pending).
4. If the QYCMCIMOM job is not running, type WRKJOB QYCMCIMOM.
5. Select the most recent job by typing 1 (Select) next to it.
6. If status is OUTQ, type 4 (Work with spooled files), and then type 5 (Display) next to the QPJOBLOG file.

Check whether CIM client request failures result in SSL certificate-related messages.

Make sure the remote CIM server certificate is added to the trust store file on the client system.

Make sure that the client's certificate and private key are being used by the client application and that the client certificate is trusted by the server.

Pegasus command-line utilities

The open source Pegasus implementation includes a set of command line utilities that you can use to control or change the Pegasus environment.

The i5/OS version of Pegasus includes these commands, with several changes to support running Pegasus on i5/OS. During normal use, administrators should rarely need to use these commands.

The commands include:

- **cimmofl**
- **cimconfig**
- **cimprovider**
- **ssltrustmgr**

For the i5/OS implementation, the **cimmofl**, **cimconfig**, and **cimprovider** commands require *IOSYSCFG and *ALLOBJ special authorities. The **ssltrustmgr** command requires *SECADM and *ALLOBJ special authorities.

All of the command-line utilities need to run from a QSHELL command-line, which requires that the QSHELL product (SS1 opt. 30) be installed on the system. You can run these commands from /QIBM/UserData/OS400/CIM.

Related concepts

“Configure Pegasus” on page 3

The Pegasus CIMOM provides the ability to set a number of configuration options.

“cimmofl usage information”

Use this command to compile provider registrations and to compile CIM class descriptions (using the Managed Object Format [MOF] language) into the class schema stored in the repository.

“cimconfig usage information” on page 23

Configure the startup options for the CIMOM with the **cimconfig** command.

“cimprovider usage information” on page 25

Enable or disable a registered provider, primarily during testing with this command. The CIMOM must be running to use this command.

cimmofl usage information

Use this command to compile provider registrations and to compile CIM class descriptions (using the Managed Object Format [MOF] language) into the class schema stored in the repository.

The CIMOM must be stopped before using this command.

Note: You can run this command from /QIBM/UserData/OS400/CIM.

Name **cimmofl** - used to compile CIM class descriptions (using the MOF language) into the class schema stored in the repository. The CIMOM must be stopped before using this command.

Synopsis

```
Usage:
cimmofl -h | --help
cimmofl --version
cimmofl [ -w ] [ -E ] [ -uc ] [ -aE | -aV | -aEV ] [ -I path ] [ -n namespace |
--namespace namespace ] [ --xml ] [ --tracefile ] [ -q ]
[ -R repositorydir | --CIMRepository repositorydir ] [ -N repositoryname ] [ -M
repositorymode] mof_file...
```

Description

This command is used to compile provider registrations and to compile CIM class descriptions (using the MOF language) into the class schema stored in the repository. The CIMOM must be stopped before using this command.

The Pegasus MOF compiler is a command-line utility that compiles MOF files (using the MOF format defined by the DMTF CIM Specification) into a Pegasus repository. It allows compiling from structures of MOF files using the include pragma and can either compile into a Pegasus

repository or perform a syntax check on the MOF files. The compiler requires that the input MOF files be in the current directory or that a fully qualified path be given. MOF files included using the #pragma include must be in the current directory or in a directory specified by a -I command-line switch.

Options

-h, --help

Print out usage message with command definitions.

--version

Displays the CIMOM server version

-E Used to perform a syntax check on the input. This option does not update the repository.

-w Suppresses warning messages.

-q Used to suppress all messages except command-line usage errors.

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

-aE Used to allow the addition or modification of classes with the experimental qualifier.

-aV Used to update 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 down leveled, 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 major version, n is minor release and u is 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 Allow both Experimental and Version Schema changes.

-R<path>

Specifies the path to the repository to be written. Specify an absolute or relative path. The default is /QIBM/UserData/OS400/CIM.

--CIMRepository<path>

If specified, this overrides the current repository path. Specify an absolute or relative path. The default is /QIBM/UserData/OS400/CIM.

-I<path>

Used to specify a path to the included MOF files.

-n<path>

Used to override the default CIM repository namespace. The default is root/cimv2.

--namespace<path>

Used to override default CIM repository namespace. The default is root/cimv2.

--xml Used to generate XML to standard output. This option does not update the repository.

--trace Used to write trace information to a file. The output destination is standard output.

--trace=<tracefile>

Used to write trace information to the specified file.

-N repositoryname

Used to specify the repository name. This is the relative path to the directory in the -R option. The default is **repository**.

-M repositorymode

Used to specify the Repository mode [XML, BIN].

Note: The default is **XML**, and this is the only allowed value on i5/OS.

Examples

```
cimmofl -w -Rtestrepository -I./myDir myDir/CIM_Schema.mof
```

Note: Compile the Managed Object Format (MOF) file located in directory `myDir` with the name `CIM_Schema.mof`. `CIM_Schema.mof` includes pragmas for other mof files that are also in the MOF directory. It will create the repository in directory `testrepository` using the default namespace `root/cimv2`. It assumes that the `testrepository` directory exists. Use the `-w` option to suppress warning messages.

```
| cimmofl -w -R/qibm/userdata/os400/cim -I/qibm/proddata/os400/cim/schemas/cim  
| -nroot/cimv2 /qibm/proddata/os400/cim/schemas/cim/CIM_Schema.mof
```

Note: Compile the MOF file located in the directory `/qibm/proddata/os400/cim/schemas/cim` with the name `CIM_Schema.mof`. `CIM_Schema.mof` includes pragmas for other mof files that are also in the `/qibm/proddata/os400/cim/schemas/cim` directory. It will create the repository in directory `/qibm/userdata/os400/cim` using namespace `root/cimv2`. It assumes that the `/qibm/userdata/os400/cim` directory exists. Use the `-w` option to suppress warning messages.

iSeries-specific usage: On an iSeries server, this command requires the user to have `*IOSYSCFG` and `*ALLOBJ` authority.

Related concepts

“Pegasus command-line utilities” on page 20

The open source Pegasus implementation includes a set of command line utilities that you can use to control or change the Pegasus environment.

cimconfig usage information

Configure the startup options for the CIMOM with the **cimconfig** command.

```
| If you change any configuration properties that are in the planned configuration settings, changes will  
| not take effect until the CIM server is restarted.
```

```
| Note: You can run this command from /QIBM/UserData/OS400/CIM.
```

Name `cimconfig - get, set, unset or list CIMOM configuration properties.`

Synopsis

Usage:

- `cimconfig -g name [-c] [-d] [-p] [-q]`
- `cimconfig -s name=value [-c] [-p] [-q]`
- `cimconfig -u name [-c] [-p] [-q]`
- `cimconfig -l [-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 allows to set the current value and planned value of the specified configuration property to the specified value.

The third form allows unsetting the current and planned values of the specified configuration property to its default value.

The last form of this command allows for all the configuration properties to be listed. Specifying the `-c` or `-p` options, will provide a listing of all the current or planned configuration property names and values.

Options

The `cimconfig` command recognizes the following options:

- | **-h, --help**
| Displays command help information
- | **--version**
| Displays CIMOM server version
- g name**
Gets the current value of the specified configuration property. Returns an error when the CIMOM is not running.
- g name -c**
Gets the current value of the specified configuration property. Returns an error when the CIMOM is not running.
- g name -p**
Gets the planned value of the specified configuration property.
- g name -d**
Gets the default value of the specified configuration property. Returns an error when the CIMOM is not running.
- s name=value**
Indicates that a configuration property is to be added or updated by setting its current value to the specified value. Returns an error when the CIMOM is not running or the specified property cannot be updated dynamically.
- s name=value -c**
Indicates that a configuration property is to be added or updated by setting its current value to the specified value. Returns an error when the CIMOM is not running or the specified property cannot be updated dynamically.
- s name=value -p**
Indicates that a configuration property is to be added or updated by setting its planned value to the specified value.
- u name**
Indicates that the current value of the specified configuration property is to be reset to the default value. Returns an error when the CIMOM is not running or the specified property cannot be updated dynamically.
- u name -c**
Indicates that the current value of the specified configuration property is to be reset to the default value. Returns an error when the CIMOM is not running or the specified property cannot be updated dynamically.
- u name -p**
Indicates that the planned value of the specified configuration property is to be reset to the default value.
- l** Displays the name of all the configuration properties. Returns an error when the CIMOM is not running.

- l -c Displays the name and value pair of all the current configuration properties. Returns an error when the CIMOM is not running.
- l -p Displays the name and value pair of all the planned configuration properties.
- q Quiet option specifies that no output is sent to standard output or standard error.

iSeries-specific usage: On an iSeries server, this command requires the user to have *IOSYSCFG and *ALLOBJ authority.

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

Related concepts

“Backup and recovery considerations” on page 17

Regularly back up the Pegasus repository as part of your existing backup plan. In most cases, you can recover a damaged repository by restoring the last backup copy.

“Pegasus command-line utilities” on page 20

The open source Pegasus implementation includes a set of command line utilities that you can use to control or change the Pegasus environment.

“Configure Pegasus” on page 3

The Pegasus CIMOM provides the ability to set a number of configuration options.

Related tasks

“Secure Pegasus” on page 12

Use this topic to find out about the options that are available for ensuring that the CIM server is secure.

cimprovider usage information

Enable or disable a registered provider, primarily during testing with this command. The CIMOM must be running to use this command.

Note: You can run this command from /QIBM/UserData/OS400/CIM.

Name cimprovider - disable, enable, remove or list registered CIM providers or CIM provider modules and module status.

Synopsis

Usage:

- cimprovider -d -m module [-q]
- cimprovider -e -m module [-q]
- cimprovider -r -m module [-p provider] [-q]
- 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

The cimprovider command provides a command-line interface to disable, enable, unregister, and list registered CIM providers. If a CIM provider is disabled, the CIMOM rejects any requests to

the provider. If a CIM provider is enabled, the CIMOM forwards requests to the provider. And if a CIM provider is unregistered, the CIMOM will no longer have any information about the provider. In order to use the `cimprovider` command, `cimserver` has to be running and the specified provider module (a grouping of providers in the same *SVRPGM) or provider has to be registered with the CIMOM.

The first form of `cimprovider` 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 will be rejected.

The second form of `cimprovider` enables the providers that are contained in the specified provider module. The providers that are contained in the specified provider module are now ready to accept new request.

The third form of `cimprovider` removes (unregisters) the specified provider module and all of its contained providers or the specified provider in the specified provider module. Once removed a provider or provider module, must be reregistered (typically by loading its registration schema using the `cimmofl` command).

The last form of `cimprovider` lists all the registered provider modules and module status or all the providers in the specified provider module. To list all providers in all modules, issue a `cimprovider -l` command, followed by `cimprovider -l -m` for each listed module.

Options

The `cimprovider` command recognizes the following options:

- | **-h, --help**
| Displays command help information.
- | **--version**
| Displays the CIMOM server version.
- | **-d** Disables the specified CIM provider module. If the module is already disabled, an error message is returned.
- | **-e** Enables the specified CIM provider module. If the module is already enabled or is currently being disabled, an error message is returned.
- | **-r** Removes the specified provider module and all of its contained providers. If a provider is specified, removes the specified provider in the specified provider module without affecting any other providers in that module.
- | **-l** Displays all the registered provider modules.
- | **-m Module**
| Specifies the provider module for the operation.
- | **-p Provider**
| Specifies the provider for the operation.
- | **-q** Quiet option specifies no output is sent to standard output or standard error.
- | **-s** Displays the status of provider modules.

Examples

cimprovider -d -m myProviderModule

Disable provider module `myProviderModule` and all of its contained providers (placing them in a stopped state).

cimprovider -e -m myProviderModule

Enable provider module `myProviderModule` and all of its contained providers (placing them in a OK state).

cimprovider -r -m myProviderModule

Remove (unregister) the myProviderModule provider module and all of its contained providers.

cimprovider -r -m myProviderModule -p MyProvider

Remove (unregister) the MyProvider provider that is contained in the myProviderModule provider module.

cimprovider -l

List the registered provider modules.

cimprovider -l -s

List the registered provider modules and their status (such as OK, Stopping, Stopped).

cimprovider -l -m myProvider

List the registered providers that are in the myProviderModule provider module.

iSeries server-specific usage: On an iSeries server, this command requires the user to have *IOSYSCFG and *ALLOBJ authority.

Related concepts

“Pegasus command-line utilities” on page 20

The open source Pegasus implementation includes a set of command line utilities that you can use to control or change the Pegasus environment.

ssltrustmgr usage information

This command provides a command-line interface to manage X.509 certificates in a trust store or a Certificate Revocation List (CRL).

You must run the ssltrustmgr command from a QSHHELL command-line, which requires that the QSHHELL product is installed on the system. You can run this command from /QIBM/UserData/OS400/CIM.

The CIMOM must be running to use this command.

Name ssltrustmgr - add, remove, revoke or list X.509 certificates in a PEM format trust store.

Synopsis

Usage:

- ssltrustmgr -a [-t truststore] -c certuser -f certfile
- ssltrustmgr -a -T trustpath -f certfile
- ssltrustmgr -a -R -f crlfile
- ssltrustmgr -r [-t truststore | -T trustpath] -i issuername -n serialnumber
- ssltrustmgr -r -R -i issuername
- ssltrustmgr -l [-t truststore | -T trustpath] [-i issuername [-n serialnumber]]
- ssltrustmgr -l -R [-i issuername]
- ssltrustmgr -h | --help
- ssltrustmgr -v | --version

Remarks

This command exits with an error status if the user running the command is not a privileged user. A privileged user has *ALLOBJ and *SECADM special authorities. The ssltrustmgr command requires that the CIM Server is running.

Description

The ssltrustmgr command provides a command-line interface to manage X.509 certificates in a trust store or a Certificate Revocation List (CRL). The command exits with an error status if the trust store or the CRL store do not exist or they are not in directory format.

The add option of the ssltrustmgr command adds an X.509 certificate from one of the following:

- The **certfile** to the specified **truststore** or **trustpath**.
- The CRL from **crfile** to the CRL store.

The **truststore** names supported are **cim_trust** and **export_trust**. If no **truststore** is specified, then **cim_trust** is used as the default **truststore**. If **truststore** is specified, then **certuser** must be specified. The **certuser** specifies the username to be associated with the certificate in the **certfile**. If the CRL specified in **crfile** already exists in the CRL store, the existing CRL is overwritten.

The remove option of the **ssltrustmgr** command removes the X.509 certificate matching the specified **issuename** and **serialnumber** from the specified **truststore** or **trustpath**. The remove option also removes the CRL from the CRL store for a specified **issuename**.

The list option of the **ssltrustmgr** command lists the X.509 certificates in the specified **truststore** or **trustpath**. The listing can be filtered by specifying the **issuename** and **serialnumber**. The list option also lists the CRLs for the specified **issuename**.

Options

The **ssltrustmgr** command uses the following options:

- a** Adds the specified certificate to the target **truststore**, **trustpath**, or a CRL store. If the **truststore** or **trustpath** does not exist, an error message is returned and no action is taken. If the specified certfile does not contain an X.509 certificate or contain an invalid certificate, an error message is returned and no action is taken. If the specified **crfile** contains an invalid CRL, an error message is returned and no action is taken. If the CRL specified in **crfile** already exists in the CRL store, the existing CRL is overwritten.
- r** Removes the certificate matching the **serialnumber** issued by the **issuename** from the target **truststore** or **trustpath**. If no certificate exists for the specified **issuename** and **serialnumber**, an error message is returned and no action is taken. If **-R** option is specified, it removes the CRL issued by the specified **issuename**.
- l** Displays the X.509 certificates in the target **truststore** or **trustpath**. If **issuename** and **serialnumber** are specified, only the matching certificates are displayed. If **-R** option is specified, all the CRLs in the CRL store are displayed. If **issuename** is specified with the **-R** option, then the CRL issued by that issuer is displayed.
- R** Indicates that the requested add, remove, or list operation is to be performed on the CRL store.
- t truststore**
Specifies a trust store name containing zero or more X.509 certificates.
- T trustpath**
Specifies a trust store path containing zero or more X.509 certificates.
- f certfile / crfile**
Specifies a PEM format file containing an X.509 certificate or a CRL.
- c certuser**
Specifies a **username** to be associated with the specified certificate. The **username** specified should be a valid system user on the target system.
- i issuename**
Specifies a certificate or a CRL issuer name.
- n serialnumber**
Specifies a certificate serial number.
- h | --help**
Displays command help information.
- v | --version**
Displays the CIMOM version number.

| **Exit status**

| When an error occurs, an error message is written to stderr and an error value 1 is returned. The following values are returned:

| 0 Success

| 1 Error

| **Examples**

| **ssltrustmgr -a -t cim_trust -c username -f cert.pem**

| Adds the X.509 certificate in the **cert.pem** file to the trust store **cim_trust** on the CIMOM and associate user **username** with the certificate.

| **ssltrustmgr -a -T /QIBM/UserData/OS400/CIM/mytruststore -f cert.pem**

| Adds the X.509 certificate in the **cert.pem** file to the trust store specified by the trust path **/QIBM/UserData/OS400/CIM/mytruststore**. User association is not required when trust path is specified.

| **ssltrustmgr -a -R -f class1crl.pem**

| **ssltrustmgr -aR -f class1crl.pem**

| Both of these examples add the CRL in **class1crl.pem** to the Certificate Revocation List on the CIMOM.

| **ssltrustmgr -r -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com" -n 01**

| Removes the certificate matching the specified **issuename** and **serialnumber** from the **cim_trust** trust store.

| **ssltrustmgr -l -t export_trust**

| Lists all the X.509 certificates in the **export_trust** trust store.

| **ssltrustmgr -l**

| **ssltrustmgr -l -t cim_trust**

| Both of these examples list all the X.509 certificates in the **cim_trust** trust store.

| **ssltrustmgr -lR -i "/C=US/ST=California/L=Cupertino/O=Smart & Secure/OU=Secure Software Division/CN=dev.admin.ss.com"**

| Lists the CRL issued by the issuer name.

| **iSeries-specific usage:** On an iSeries server, this command requires the user to have *SECADM and *ALLOBJ authority.

| **Related concepts**

| "Advanced startup options for the cimconfig command" on page 5

| You can change the advanced startup options for the CIM server with the cimconfig command.

Pegasus developer's reference

The CIM standard provides the ability to develop management application that work with the systems management data exposed by the CIM providers included with i5/OS.

To work with CIM, developers should have a thorough understanding of the CIM standard defined by the DMTF. For more information about the CIM standard, see CIM Specification 2.2 on the Distributed Management Task Force (DMTF) Web site.

Pegasus on i5/OS does not support Service Location Protocol (SLP).

Pegasus also includes development tools, samples and reference material.

IBM has included providers with i5/OS that support basic operating system information and some performance metrics.

Related concepts

“i5/OS metrics” on page 39

i5/OS supports the Common Management Model (CMM)

Related information

CIM Specification 2.2

Pegasus Web site

Supported CIM base operating system classes

IBM has implemented CIM classes as IBM-supplied providers to provide basic operating system information.

The providers supplied by IBM are as follows:

- IBMOS400_ComputerSystem: subclass of CIM_Computer_System
- IBMOS400_OperatingSystem: subclass of CIM_OperatingSystem
- IBMOS400_RunningOS: subclass of CIM_RunningOS
- IBMOS400_Process: subclass of CIM_Process
- IBMOS400_OSProcess: subclass of CIM_OSProcess
- | • IBMOS400_VirtualProcessor: subclass of CIM_Processor
- | • IBM_IPProtocolEndpoint: a subclass of CIM_IPProtocolEndpoint
- | • IBM_LocalFileSystem: a subclass of CIM_LocalFileSystem
- | • IBM_RemoteFileSystem: a subclass of CIM_RemoteFileSystem
- | • IBM_NFS: a subclass of CIM_NFS
- | • IBMOS400_NetworkPort: a subclass of CIM_NetworkPort
- | • IBM_EthernetPort: a subclass of CIM_EthernetPort
- | • IBM_TokenRingPort: a subclass of CIM_TokenRingPort
- | • IBMOS400_CSVirtualProcessor: a subclass of CIM_SystemDevice
- | • IBM_CSNetworkPort: a subclass of CIM_SystemDevice
- | • IBMOS400_HostedFileSystem: a subclass of CIM_HostedFileSystem
- | • IBM_BootOSFromFS: a subclass of CIM_BootOSFromFS
- | • IBM_NWPortImplementsIPEndpoint: a subclass of CIM_PortImplementsEndpoint

Some property values are available in several languages to CIM clients that follow the globalization interface as described in the DMTF standards.

The following figure shows the CIM base classes that are extended by the IBM extension classes.

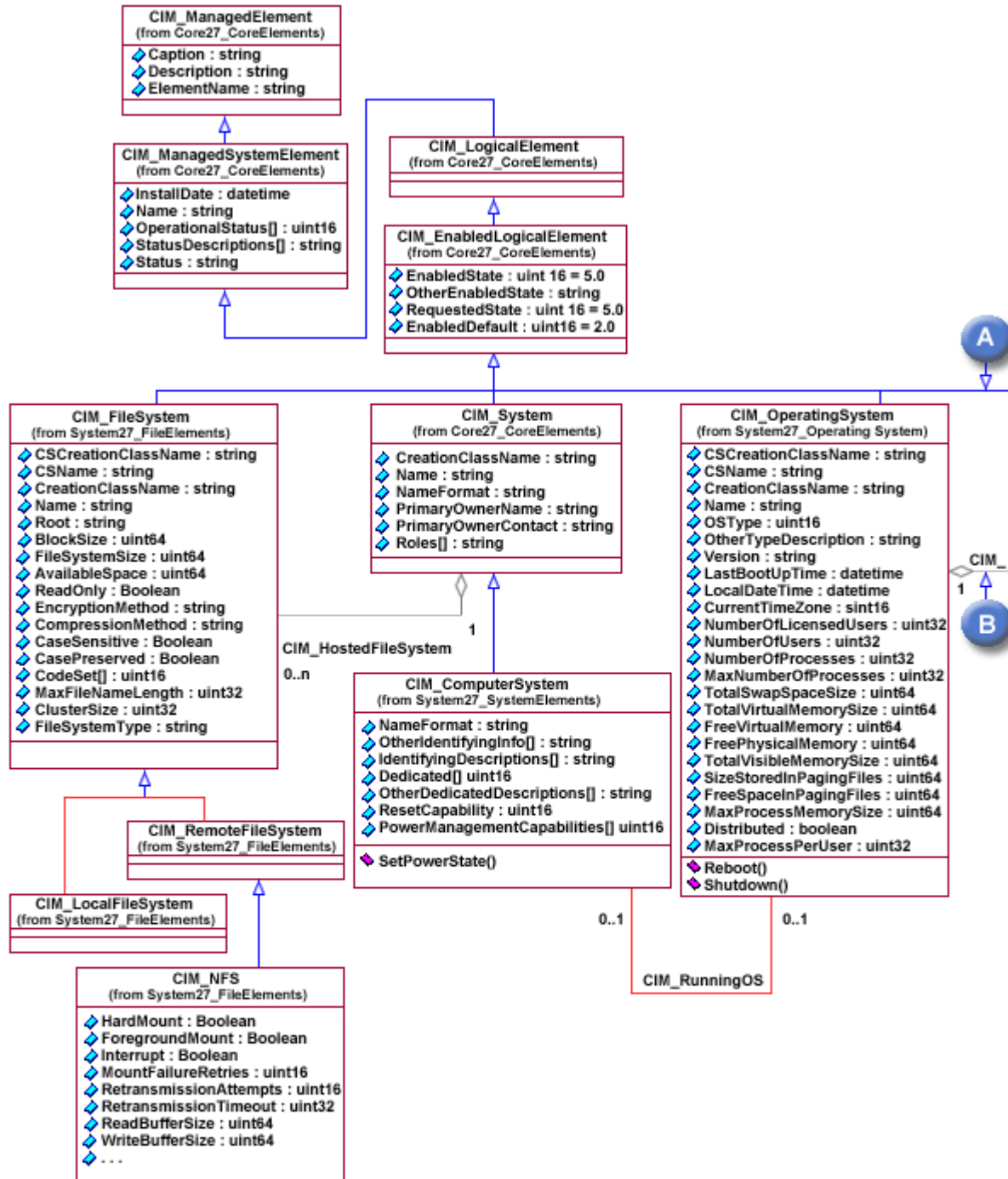


Figure 1. CIM Classes

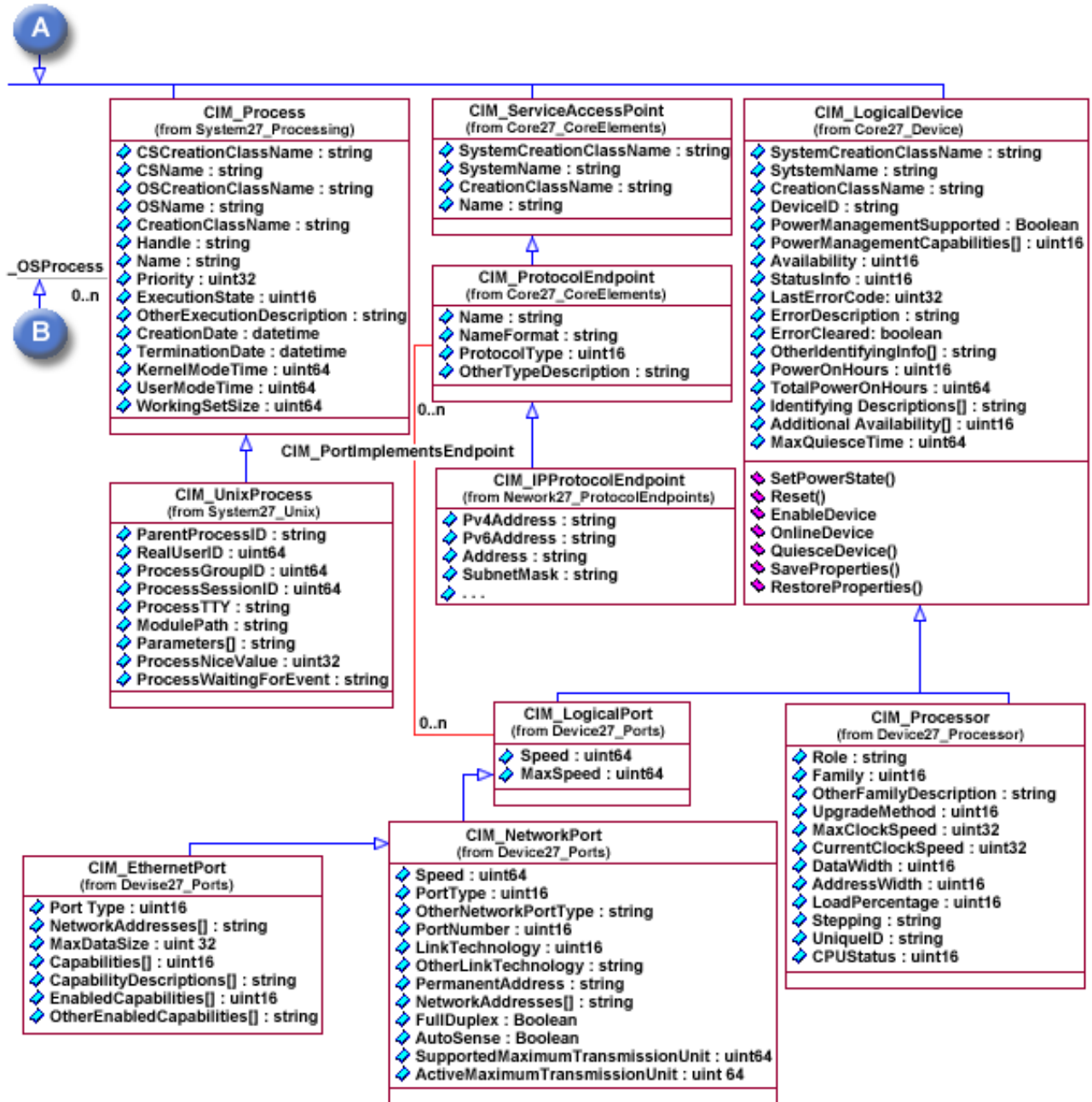


Figure 2. CIM Classes--continued

The managed object format (MOF) files that define these classes can be found in the /QIBM/ProdData/OS400/CIM/Schemas/OS400/ directory. The file names are as follows:

- IBMOS400_OSBase.mof
- IBMOS400_BootOSFromFS.mof
- IBMOS400_CSNetworkPort.mof
- IBMOS400_CSVirtualProcessor.mof
- IBMOS400_FileSystem.mof
- IBMOS400_HostedFileSystem.mof
- IBMOS400_NWPortImplementsIPEndpoint.mof
- IBMOS400_NetworkPort.mof

- IBMOS400_ProtocolEndpoints.mof
- IBMOS400_VirtualProcessor.mof

IBMOS400_ComputerSystem

This class makes available basic computer system information such as computer name, and status information. This provider instrumenting this class is for use by clients as part of a basic understanding of the identity of the managed system on which it is running (typically a server or appliance).

The following properties have data that can be specific to i5/OS, or can map to i5/OS system-specific attributes.

Property	Property value and data source
string OtherIdentifyingInfo[]	Returns the system: <ul style="list-style-type: none"> • Type • Serial number • Model • Partition identifier
Name	The system name based on the first entry in the TCP/IP host table.

Note: The methods on IBMOS400_ComputerSystem have not been implemented. This provider does not register as a method provider.

IBMOS400_OperatingSystem

This class is for use by clients as part of a basic understanding of the identity of the Managed System on which the corresponding provider is running (typically a server or appliance).

The following properties have data that may be specific to i5/OS, or may map to i5/OS specific attributes.

Property	Property value and data source
uint32 NumberOfUsers	The number of interactive jobs on the system
uint32 NumberOfProcesses	The total number of user jobs and system jobs that are currently in the system. The total includes: <ul style="list-style-type: none"> • All jobs on job queues waiting to be processed. • All jobs currently active (being processed). • All jobs that have finished running but still have output on output queues to be produced.
uint32 MaxNumberOfProcesses	The maximum number of jobs that are allowed on the system.
string LanguageEdition	Returns QLANGID system value.
string CodeSet (maximum length is 64)	Returns code page field from QCHRID system value.
uint32 DefaultPageSize (in bytes)	Always returns 4096.

| **Note:** For the IBMOS400_OperatingSystem class, the Shutdown() and Reboot() methods are implemented
| in V5R4.

IBMOS400_RunningOS

This provider for this class is for use by clients to find associations between a computer system and the operating system that is currently running on the computer system.

IBMOS400_Process

The provider for this class makes available basic process information such as process name (a fully qualified job name on i5/OS), priority, runtime state. An IBMOS400_Process is mapped to i5/OS jobs on the iSeries server. Client applications can use this provider to give clients an understanding of the processes (jobs) running on the managed system within the context of its operating system.

Note: There are many (potentially thousands) of i5/OS jobs running or in queues on any given i5/OS system. For performance reasons, any CIM client application that needs to obtain a list of i5/OS jobs should first EnumerateInstanceNames and then perform a GetInstance for each i5/OS job for which instance data is required.

The following properties have data that can be specific to i5/OS, or can map to i5/OS specific attributes.

Property	Property value and data source
string ElementName	Returns string from the Name field below - the fully-qualified job name on i5/OS as viewed on command-line interface.
string Name	This is the fully-qualified job name on i5/OS as viewed on the command-line interface: <i>Job number</i> field + / + <i>User name</i> field + / + <i>Job name</i> field where <i>Job number</i> is the system-generated job number. <i>User name</i> is the user name of the job, which is the same as the name of the user profile under which the job was started. <i>Job name</i> is the name of the job as identified to the system.
string Handle [Key]	This is the fully-qualified job name on the i5/OS as used in the i5/OS work management APIs: <i>Job name</i> field + / + <i>User name</i> field + / + <i>job number</i> field where <i>Job name</i> is the name of the job as identified to the system. <i>User name</i> is the user name of the job, which is the same as the name of the user profile under which the job was started. <i>Job number</i> is the system-generated job number.
uint16 ExecutionState	Based on the Job Status obtained from Retrieve Job Information (QUSRJOBI) API, Format JOBI0100, Job status field. This is based on the i5/OS active job status. The i5/OS active job status values are mapped to the ExecutionState values as follows:

Property	Property value and data source	
	No status	0 ("Unknown")
	BSCA	4 ("Blocked")
	BSCW	4 ("Blocked")
	CMNA	4 ("Blocked")
	CMNW	4 ("Blocked")
	CMTW	4 ("Blocked")
	CNDW	4 ("Blocked")
	CPCW	4 ("Blocked")
	DEQA	4 ("Blocked")
	DEQW	4 ("Blocked")
	DKTA	4 ("Blocked")
	DKTW	4 ("Blocked")
	DLYW	6 ("Suspended Ready")
	DSC	1 ("Other")
	DSPA	5 ("Suspended Blocked")
	DSPW	5 ("Suspended Blocked")
	END	1 ("Other")
	EOFA	4 ("Blocked")
	EOFW	4 ("Blocked")
	EOJ	1 ("Other")
	EVTW	5 ("Suspended Blocked")
	GRP	6 ("Suspended Ready")
	HLD	6 ("Suspended Ready")
	HLDT	6 ("Suspended Ready")
	ICFA	4 ("Blocked")
	ICFW	4 ("Blocked")
	INEL	1 ("Other")
	JVAA	4 ("Blocked")
	JVAW	4 ("Blocked")
	LCKW	5 ("Suspended Blocked")
	LSPA	4 ("Blocked")
	LSPW	4 ("Blocked")
	MLTA	4 ("Blocked")
	MLTW	4 ("Blocked")
	MSGW	5 ("Suspended Blocked")
	MTXW	5 ("Suspended Blocked")
	MXDW	4 ("Blocked")
	OPTA	4 ("Blocked")
	OPTW	4 ("Blocked")
	OSIW	4 ("Blocked")
	PRTA	4 ("Blocked")
	PRTW	4 ("Blocked")
	PSRW	2 ("Ready")
	RUN	3 ("Running")
	SELW	4 ("Blocked")
	SEMW	5 ("Suspended Blocked")
	SIGS	8 ("Suspended Blocked")
	SIGW	5 ("Suspended Blocked")
	SRQ	6 ("Suspended Ready")
	SVFA	4 ("Blocked")
	SVFW	4 ("Blocked")
	TAPA	4 ("Blocked")
	TAPW	4 ("Blocked")
	THDW	4 ("Blocked")
	TIMA	6 ("Suspended Ready")
	TIMW	6 ("Suspended Ready")
	All other values	0 ("Unknown")

Property	Property value and data source
string OtherExecutionDescription	Returns the active job status as a string. Because most of the active job status take an entire sentence to explain, the CHAR(4) active job status itself is returned.
datetime CreationDate	Date and time job became active when the job began to run on the system. This information is not available if the job did not become active.
datetime TerminationDate	Date and time job ended. When the job running on the system was complete. If the job was not complete, zeros are returned.
uint64 KernelModeTime (in milliseconds)	System time spent running. Because i5/OS does not distinguish between system time and user time, this value cannot be reported. Returns 0 (zero) indicating information is not available.
uint64 UserModeTime (in milliseconds)	User time spent executing. Since i5/OS does not distinguish between system time and user time, this value reflects the processing unit time (in milliseconds) that the job used. This information is available only for active jobs.

IBMOS400_OSProcess

The provider for this class provides a link between the operating system and processes running in the context of this operating system. Client applications can use this provider to give clients an understanding of the processes (jobs) running on the managed system within the context of its operating system.

IBMOS400_VirtualProcessor

The provider for this class models an internal hypervisor array element that represents an internal virtual processor for the partition.

Note: Only active processors are returned.

Property	Property value and data source
string DeviceID [Key]	This number represents an index into an internal array in the hypervisor. This number is converted to a string.
uint16 OperationalStatus[] (ValueMap)	Indicates the current status of the element. Always returns 2 (OK).

IBM_IPProtocolEndpoint properties

An IBM_IPProtocol Endpoint is mapped to a TCP interface on an i5/OS iSeries system.

Property	Property value and data source
string Description	The IP address and the associated line description. For example, The i5/OS IP protocol endpoint named 1.2.3.4, and associated with the line description TRNLINE.
string ElementName	The TCP interface IP address. For example 1.2.3.4.

IBM_LocalFileSystem, IBM_RemoteFileSystem, IBM_NFS Properties

The IBM_LocalFileSystem class models the Root, QOpenSys, QDLS, QSYSLIB, UDFS, Optical and IASP QSYSLIB local file systems on i5/OS. The IBM_RemoteFileSystem class models the QFileServer400, Netware and QNTC remote file systems for i5/OS. The IBM_NFS class models the NFS file system for i5/OS.

Property	Property value and data source
string ElementName	Indicates the path name or other information defining the root of the file system.
string Name	Indicates the key of a file system instance within a computer system.
string Root	Indicates the path name or other information defining the root of the file system.

IBMOS400_NetworkPort, IBM_EthernetPort, and IBM_TokenRingPort Properties

The **IBM_EthernetPort** property models the Ethernet line descriptions for i5/OS. The **IBM_TokenRingPort** property models the token ring line descriptions for i5/OS. The **IBMOS400_NetworkPort** property models all other line description types.

Property	Property value and data source
string Description	The name of the line description and its type.
string ElementName	The name of the line description.

IBMOS400_CSVirtualProcessor

IBMOS400_CSVirtualProcessor is an association class that associates a partition virtual-processor with a computer system.

Property	Property value and data source
IBMOS400_ComputerSystem REF GroupComponent	The parent computer system in the association. Returns a reference to the IBMOS400_ComputerSystem class.
IBMOS400_VirtualProcessor REF PartComponent	The virtual processor that is a component of a computer system. Returns a reference to the IBMOS400_VirtualProcessor class representing this particular virtual processor.

IBM_CSNetworkPort

IBM_CSNetworkPort is an association class that associates a network port with a computer system.

Property	Property value and data source
IBMOS400_ComputerSystem REF GroupComponent	The parent computer system in the association returns a reference to the IBMOS400_ComputerSystemIBM class. Returns a reference to the IBMOS400_ComputerSystem class.
CIM_NetworkPort REF PartComponent	The network port that is a component of a computer system. Returns a reference to the CIM_NetworkPort class representing this particular network port.

IBMOS400_HostedFileSystem

The IBMOS400_HostedFileSystem association class associates a file system with a computer system.

Property	Property value and data source
IBMOS400_ComputerSystem REF GroupComponent	The parent computer system in the association. Returns a reference to the IBMOS400_ComputerSystem class.

Property	Property value and data source
CIM_FileSystem REF PartComponent	The file system that is a component of a computer system. Returns a reference to the CIM_FileSystem class representing this particular file system.

IBM_BootOSFromFS

The IBM_BootOSFromFS association class associates a file system with an operating system.

Property	Property value and data source
CIM_FileSystem REF Antecedent	The file system from which the operating system is loaded. Returns a reference to the CIM_FileSystem class.
IBMOS400_OperatingSystemREF Dependent	The operating system. Returns a reference to the IBMOS400_OperatingSystem class.

IBMOS400_NWPortImplementsIPEndpoint

The IBMOS400_NWPortImplementsIPEndpoint association class associates a logical port with a protocol endpoint.

Property	Property value and data source
CIM_NetworkPort REF Antecedent	The network port that represents the device behind the IP protocol endpoint. Returns a reference to the CIM_NetworkPort class.
IBM_IPProtocolEndpoint REF Dependent	The IPProtocolEndpoint implemented on the logical port. Returns a reference to the IBM_IPProtocolEndpoint class.

Supported CIM metric classes

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

- IBMOS400_ColSrvMetricDefinition: a subclass of CIM_BaseMetricDefinition
- IBMOS400_ColSrvMetricValue: a subclass of CIM_BaseMetricValue
- IBMOS400_ColSrvMetricInstance: a subclass of CIM_MetricInstance - association between metric definition and metric value classes or instances.
- IBMOS400_ColSrvMetricDefForME: a subclass of CIM_MetricDefForME - association between a managed element (resource) and metric definition class or instances.
- IBMOS400_ColSrvMetricForME: a subclass of CIM_MetricForME - association between a managed element (resource) and metric value class or instances.

Note: All instances of IBMOS400_ColSrvMetricValue return volatile data; only current data is supported. Historical data is not supported this release.

For a list of the metrics supported in i5/OS, see i5/OS metrics. Also see the CIM class and instance MOF files. The class MOF file, IBMOS400_ColSrvMetric.mof, and the instance MOF file, IBMOS400_ColSrvMetricDefInstance.mof can be found in /QIBM/ProdData/OS400/CIM/Schemas/OS400/.

The following figure illustrates the relationship between the IBM extension classes, and the CIM base classes they extend:

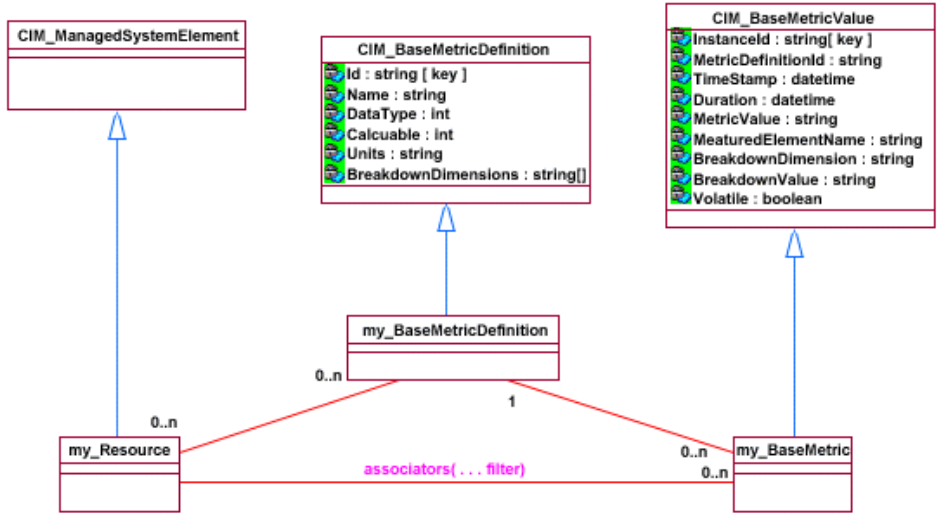


Figure 3. CIM Classes extended by i5/OS metric classes

i5/OS metrics

i5/OS supports the Common Management Model (CMM)

Table 1 describes the CMM metrics i5/OS supports:

Table 1.

Resource/Base CIM class	Metric
IBMOS400_OperatingSystem	NumberOfUsers Number of interactive jobs active during the sample interval
	NumberOfProcesses Number of jobs active during the sample interval
	FreeVirtualMemory Free space in system Auxiliary Storage Pool (ASP)
	FreePhysicalMemory i5/OS always returns 0
	FreeSpaceInPagingFiles Free space in system ASP
	PageInRate Number of pages paged in per second in all pools
	OperationalStatus i5/OS always returns 0K
	UserModeTime Total system CPU time used
	KernelModeTime i5/OS always returns 0
	TotalCPUTime Same as UserModeTime
	Internal view kernel mode percentage i5/OS always returns 0
	InternalViewUserModePercentage i5/OS always returns 0
	InternalViewTotalCPUPercentage User mode percentage as seen from within the operating system
	InternalViewIdlePercentage Idle percentage as seen from within the operating system
	CPUConsumptionIndex CPU time used divided by CPU time which might have been used by this operating system.
	ExternalViewTotalCPUPercentage External view CPU percentage
	ExternalViewUserModePercentage External view user mode percentage
ExternalViewKernelModePercentage i5/OS always returns 0	

Table 1. (continued)

Resource/Base CIM class	Metric
IBMOS400_ComputerSystem	<p>PctPartitionDefinedCapacityUsed System CPU time used as a percentage of configured capacity (the amount of CPU the logical partition is configured to use)</p> <p>ActiveVirtualProcessors Average number of virtual processors active.</p> <p>UnusedPartitionCPUCapacity Reserved but unused capacity for this OS container</p> <p>UnusedGlobalCPUCapacity CPU time in milliseconds not used on global server level</p>
IBMOS400_Process	<p>KernelModeTime i5/OS always returns 0</p> <p>UserModeTime The CPU time used by the JOB (including all secondary threads)</p> <p>TotalCPUTime Same as UserModeTime</p> <p>InternalViewKernelModePercentage i5/OS always returns 0</p> <p>InternalViewUserModePercentage Percentage value related to UserModeTime, the percentage the system CPUs were used for this process in user mode during the measurement interval.</p> <p>InternalViewTotalCPUPercentage Percentage value related to TotalCPUTime</p> <p>ExternalViewTotalCPUPercentage External view total CPU percentage</p> <p>ExternalViewUserModePercentage External view user mode percentage</p> <p>ExternalViewKernelModePercentage i5/OS always returns 0</p> <p>AccumulatedKernelModeTime i5/OS always returns 0</p> <p>AccumulatedUserModeTime CPU time in user mode spent for this process since process creation</p> <p>AccumulatedTotalCPUTime CPU time spent for this process since process creation</p>
IBMOS400_Processor	<p>TotalCPUTimePercentage The time a virtual processor was used as a percentage of the elapsed interval time.</p>
IBMOS400_NetworkPort	<p>BytesTransmitted The total number of bytes transmitted, including framing characters.</p> <p>BytesReceived The total number of bytes received, including framing characters.</p> <p>ErrorRate Number of network errors per second</p>

Related concepts

“Pegasus developer’s reference” on page 29

The CIM standard provides the ability to develop management application that work with the systems management data exposed by the CIM providers included with i5/OS.

| i5/OS support for the CIM indication provider

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

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

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

| The QYCP_ColSrvMetricIndicationProvider C++ class is the main class involved in handling indications for metrics. The indication provider defines the following methods:

CIM provider types and operations	Description
enableIndications	Called when the provider is expected to begin generating indications.
createSubscription	Tells the provider to monitor for indications matching the specified subscription.
modifySubscription	Informs the provider that the specified subscription instance has changed.
deleteSubscription	Informs the provider to stop monitoring for indications matching the specified subscription.
disableIndications	Tells the provider not to generate any more indications.

| Related information

| Open Group CIMIndicationProvider documentation

Related information for CIM

Listed here are the Web sites and information center topics that relate to the Common Information Model topic.

Web sites

- WBEM Web site(<http://www.dmtf.org/standards/wbem>)
The site is the official home of the Web-Based Enterprise Management (WBEM) Initiative.
- Introduction to CIM(<http://www.wbemsolutions.com/tutorials/CIM/cim.html>)
This site provides a tutorial of CIM.
- Pegasus Web site(<http://www.openpegasus.org>)
This is the OpenPegasus home page.
- OpenSSL documentation(<http://www.openpegasus.org>)
This site provides documentation for OpenSSL.

Other information

- Network Authentication Service topic
- Host name resolution considerations topic
- Manage keytab files topic
- Back up your server topic
- Digital Certificate Manager topic
- CIM topic in the IBM Systems Software Information Center

Saving PDF files

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

1. Right-click the PDF in your browser (right-click the link above).
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

- 1 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).

Appendix. Notices

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

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

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

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

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

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

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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

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

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

Copyright (c) 2000, 2001, 2002 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; The Open Group; Tivoli Systems. Copyright (c) 2003 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; EMC Corporation; The Open Group. Copyright (c) 2004 BMC Software; Hewlett-Packard Development Company, L. P.; IBM Corp.; EMC Corporation; VERITAS Software Corporation; The Open Group. Copyright (c) 2005 Hewlett-Packard Development Company, L.P.; IBM Corp. EMC Corporation; VERITAS Software Corporation; The Open Group.

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

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

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.

COPYRIGHT LICENSE:

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

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

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

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

Programming Interface Information

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

Trademarks

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

- | i5/OS
- | IBM
- | IBM logo
- | iSeries

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

Terms and conditions

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

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

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

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

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

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

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



Printed in USA