

z/OS



Common Information Model User's Guide

z/OS



Common Information Model User's Guide

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 57.

First Edition, September 2005

This edition applies to Version 1 Release 7 of z/OS (5694-A01), z/OS.e Version 1 Release 7 (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Str. 220
D-71032 Böblingen
Federal Republic of Germany

If you prefer to send comments electronically, use one of the following methods:

FAX: Your International Access Code +49+7031+16+3545
Internet: s390id@de.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 2005. All rights reserved.

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

Contents

About this document	v
Who should use this document	v
Legend for graphics showing class structures	v

Chapter 1. Before you can work with the CIM server 1

Software and hardware prerequisites	1
Installation	1
Installation directories	1
Specifying priority for z/OS CIM	2

Chapter 2. Introduction to the Common Information Model 3

Chapter 3. How to work with the CIM server 7

Setting up the CIM server	7
Customizing the CIM server	7
Setting up security for the CIM server	9
Configuring the CIM server	14
Starting and stopping the CIM server	14
CIM server configuration	15
Changing current configuration properties	15
Changing planned configuration properties	15
CIM server configuration properties	15
CIM server command-line utilities	18
cimmo/cimmo	19
cimconfig	20
cimprovider	22
Supported CIM operations	23
Sample client applications	24
Troubleshooting	25
Problem determination	25
Error messages	25

Chapter 4. z/OS Management Instrumentation for CIM 29

OS management Base classes	30
CIM_ComputerSystem	31
IBMzOS_ComputerSystem	31
CIM_OperatingSystem	32
IBMzOS_OperatingSystem	32
CIM_RunningOS	33
IBMzOS_RunningOS	33
CIM_Process	33
IBMzOS_Process	33

IBMzOS_UnixProcess	34
CIM_OSProcess	35
IBMzOS_OSProcess	35
OS management Processor classes	36
CIM_SystemDevice association	36
CIM_Processor	36
IBMzOS_Processor	37
OS management File System classes	38
CIM_HostedFileSystem	38
CIM_LocalFileSystem	38
IBMzOS_UnixLocalFileSystem	39
CIM_RemoteFileSystem	39
OS management Network classes	40
CIM_EthernetPort	41
IBMzOS_EthernetPort	42
CIM_PortImplementsEndpoint	42
CIM_IPProtocolEndpoint	42
IBMzOS_IPProtocolEndpoint	43
CIM_SystemDevice association	43

Chapter 5. Messages and logs 45

Logging	45
Checking the CIM server logs	45
CIM server messages	45
z/OS-specific messages	46

Appendix A. Sample CIM request 49

Appendix B. Classes and their providers. 53

OS management classes	53
OS monitoring classes	53

Appendix C. Accessibility 55

Using assistive technologies	55
Keyboard navigation of the user interface	55
z/OS information	55

Notices 57

Programming Interface Information	58
Trademarks	58
Terms and conditions for downloading and printing publications	59

Index 61

About this document

This document describes the implementation of the Common Information Model (CIM) and Web Base Enterprise Management (WBEM) standards for z/OS. It explains how to set up and use the CIM server and CIM resource instrumentation provided together with the z/OS operating system. The Common Information Model (CIM) is a standard data model for describing and accessing systems management data in heterogeneous environments. It allows system administrators and vendors to write applications that measure system resources in a network with different operating systems and hardware.

The focus of this document is on the z/OS-specific implementation of CIM. For more detailed information on the CIM and WBEM standards please review the information provided by the Distributed Management Task Force, which is found in the internet on the WBEM Web site.

Following are the major topics of this document:

- Chapter 3, “How to work with the CIM server,” on page 7 describes how to set up, customize and use the CIM server for z/OS.
- Chapter 4, “z/OS Management Instrumentation for CIM,” on page 29 describes the overall CIM data model provided for IBM eServer operating systems. The focus of this chapter are the z/OS-specific supplements and differences as well as how to use this information on z/OS.

Who should use this document

This document is intended for the system programmer and performance analyst responsible for measuring and improving system performance. The CIM data model described in this document can be used in writing applications that monitor system resources, for example to discover bottlenecks or to control thresholds and to provide information on which tuning actions should be applied.

Also, administrators of the z/OS CIM runtime environment should read Chapter 3, “How to work with the CIM server,” on page 7, which contains information on how to set up security for the CIM server.

Because the CIM data model provided for z/OS is used in applications that measure z/OS system performance, this document assumes that the reader has extensive knowledge of the z/OS system.

Legend for graphics showing class structures

The graphics in this book showing class structures illustrate the CIM object modeling using the UML syntax:

Construct	Description	Syntax
association	A relationship between two or more classifiers that involves connections among their instances.	_____
aggregation	A special form of association that specifies a whole-part relationship between the aggregate (whole) and the component part.	◊_____
inheritance	A relationship among classes where one class shares the structure and/or behavior defined for one or more other classes. Inheritance is the mechanism that makes generalization, subclasses, and superclasses possible.	↑

Figure 1. Object modeling with UML

Chapter 1. Before you can work with the CIM server

This chapter handles the following topics:

- “Software and hardware prerequisites”
- “Installation”
- “Specifying priority for z/OS CIM” on page 2

Software and hardware prerequisites

z/OS CIM is delivered with the z/OS operating system and runs on any hardware that can exploit the current z/OS version as described in the *z/OS Program Directory*.

Installation

You install z/OS CIM using SMP/E as described in the *z/OS Program Directory*. For additional setup steps, refer to “Setting up the CIM server” on page 7. Please note that other z/OS components that provide CIM instrumentation, like for example RMF, may require additional configuration steps which are described in the respective product documentation.

Installation directories

After a successful SMP/E installation, the components of z/OS CIM are located in the following HFS directory structure:

Table 1. Default installation directories for z/OS CIM

Directory	Description
/usr/lpp/wbem	Base HFS directory
/usr/lpp/wbem/bin	CIM server executables
/usr/lpp/wbem/lib	CIM server libraries (dlls)
/usr/lpp/wbem/install	Sample profile
/usr/lpp/wbem/provider	CIM Provider libraries
/usr/lpp/wbem/msg	CIM server message files for NLS
/usr/lpp/wbem/schemas	CIM MOF Schema files (CIM 2.8)

Table 1. Default installation directories for z/OS CIM (continued)

Directory	Description
/usr/lpp/wbem/repository	Master copy of data repository
/etc/wbem	CIM server configuration files
/var/wbem/logs	CIM server log files
/var/wbem/repository	CIM server data repository

Specifying priority for z/OS CIM

z/OS CIM does not need to run with the same high priority as other system address spaces. IBM does not recommend to put z/OS CIM into WLM service class SYSSTC. The dispatch priority should not be higher than that of important interactive workload. From general characteristics, it is interactive infrastructure workload. Depending on the specific installation, it is probably a good idea to keep it in a service class with a dispatch priority slightly higher than batch workload.

Chapter 2. Introduction to the Common Information Model

The Common Information Model (CIM) is a standard data model 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.

Figure 2 shows a sample environment in which management applications can run that use the DMTF CIM standard data model.

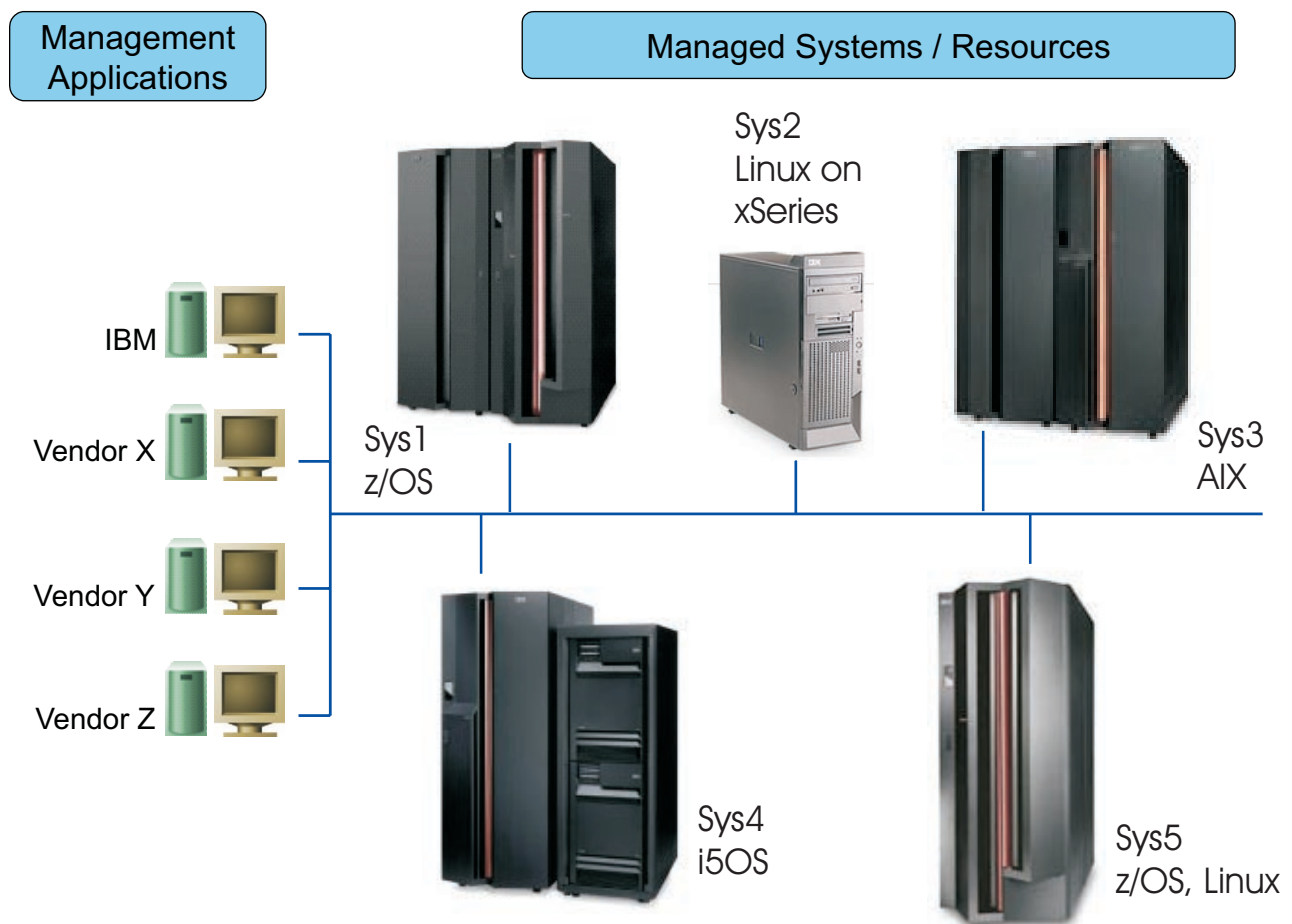


Figure 2. Sample network environment managed with CIM management applications

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

With support for the CIM server on systems running z/OS, users have the ability to access z/OS resources through an extendible industry standard model. The remainder of this document contains information on how to use the CIM server for z/OS for this purpose.

CIM for z/OS 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 CIM server.
- An open source implementation of the CIM server that manages communication between clients and providers. The CIM server also provides several management functions, including security, and a set of commands that provide configuration and management functions to administrators.

The CIM server implementation on z/OS is based on the **OpenPegasus CIM server 2.4.1** from **The Open Group**. See the OpenPegasus web site for more information (Copyright 2005. The Open Group. All Rights Reserved).

- A CIM schema that defines an information model for representing systems management functions. For z/OS 1.7, CIM Schema version 2.8 is supported by the CIM server.
- An implementation of the standardized formats for communication between clients and the CIM server, called CIM in XML, and called CIM Operations over HTTP. For more information about these standards, see the WBEM web site.
- The CIM server for z/OS 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 z/OS, refer to Chapter 4, “z/OS Management Instrumentation for CIM,” on page 29.

Figure 3 on page 5 illustrates how the CIM server works in the z/OS environment: A CIM client application requests the CIM server to return information about z/OS resources, in this case about basic OS data as well as RMF metrics. The CIM server invokes the according z/OS OS Management and RMF monitoring providers which retrieve the requested data associated to z/OS system resources. The z/OS RMF monitoring provider invokes the RMF Distributed Data Server (DDS) which in turn collects RMF Monitor III performance data. The CIM server consolidates the data from the providers and returns them back to the calling client through the CIM/XML protocol.

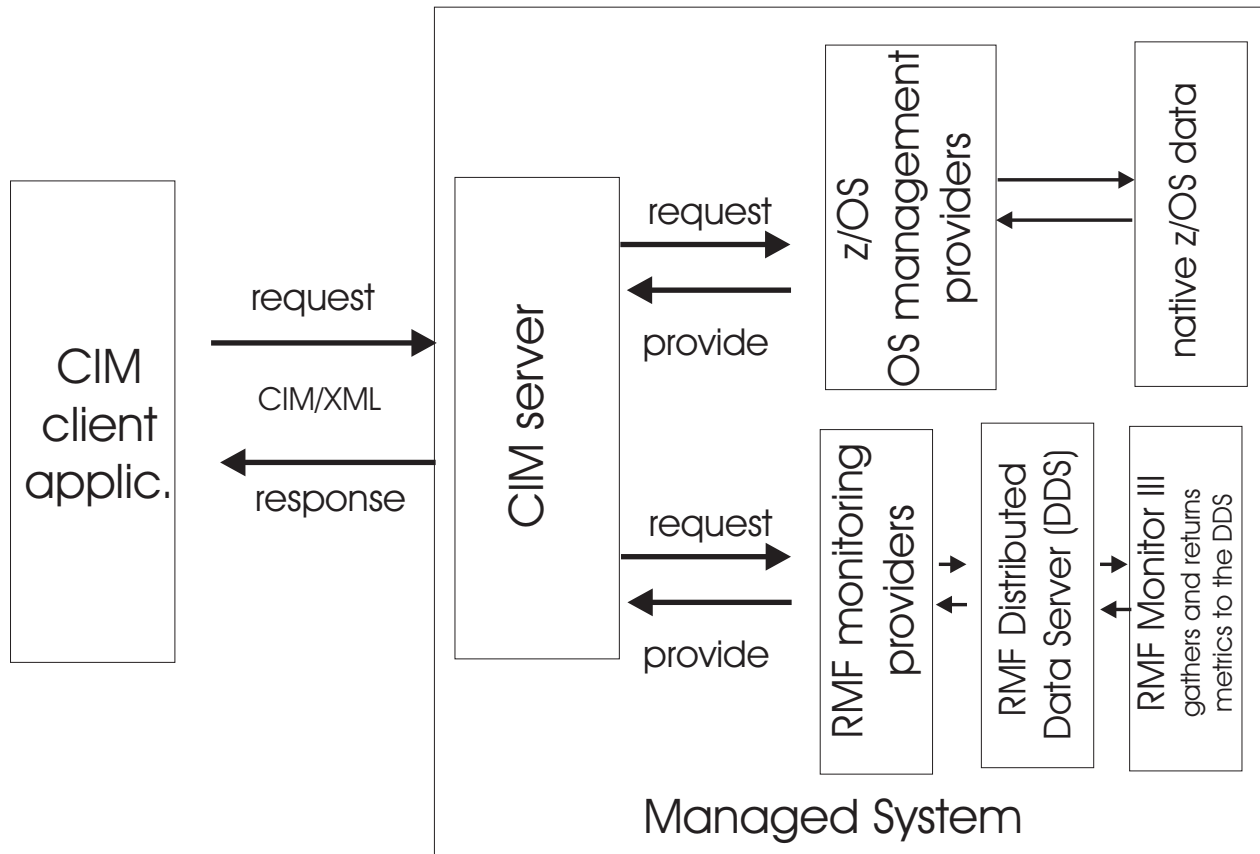


Figure 3. Tasks of the CIM server in a z/OS environment

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

Important Note:

Each IBM® @server™ operating system is supporting a specific open source implementation of a CIM server. The @server *Common Information Model* document contains overall information about how to use CIM for systems management on IBM @servers. Users of CIM for z/OS need to know this information. The present *z/OS Common Information Model User's Guide* contains the z/OS-specific supplements and deviations from the common @server CIM and from OpenPegasus.

Chapter 3. How to work with the CIM server

The CIM server is used to receive the clients' requests, collect the requested metrics/data from the managed system and return the results to the client. This chapter contains all information required for working with the CIM server. It describes the following topics:

- "Setting up the CIM server"
- "Configuring the CIM server" on page 14
- "CIM server command-line utilities" on page 18
- "Troubleshooting" on page 25

While you must set up the CIM server only once to make it ready to use, you can configure your CIM server environment as often as you want during operation to best meet your requirements. To help you in configuring the CIM server you can use the commands described in "CIM server command-line utilities" on page 18. If you run into problems while setting up or using the CIM server you can find information for problem solving in "Troubleshooting" on page 25.

Setting up the CIM server

This chapter describes the actions to be completed before the first invocation of CIM server:

- "Customizing the CIM server"
- "Setting up security for the CIM server" on page 9

Customizing the CIM server

There are two ways to start the CIM server: either from the started task procedure CFZCIM or from within a Unix System Services shell. If you want to start the CIM server as started task, you must customize the JCL procedure CFZCIM and the according environment variable file */etc/wbem/cimserver.env*.

If you want to start the CIM server from a UNIX System Services shell or a remote UNIX session (telnet, ssh), the file *profile.add* contains the environment variables that you need to define. How to perform these tasks is described in the following subsections.

Customizing started task procedure CFZCIM

You can start the CIM server via started task procedure CFZCIM if you include CFZCIM in your PROCLIB concatenation. A sample of CFZCIM is shipped with the default PROCLIB. If you have installed the CIM server below a -PathPrefix- on your target system, the PARM parameter of the EXEC statement should be updated to point to your installation directory in the HFS.

The DDNAMEs STDOUT and STDERR are used to redirect the output from the console into the HFS-files *cimserver.out* and *cimserver.err*. If the started task is ended, the console output will be copied to the JCL job log in job steps two and three.

The DDNAME STDENV points to the HFS file containing environment variables required to run the CIM server.

For running the CIM server as a started task, the environment variables are set in file *cimserver.env* located in the */etc/wbem* HFS directory. See “Customizing environment variables” for details on how to set environment variables for the z/OS CIM server.

To run the CIM server with a user ID for which the security setup has been completed, either set up the STARTED class or use the started procedures table (ICHRIN03). For further details refer to *z/OS Security Server RACF Security Administrator's Guide*, chapter *Assigning RACF User IDs to Started Procedures*.

Example of the RACF commands required to setup the CIM server for the STARTED class:

```
RDEFINE STARTED CFZCIM.* STDATA(USER(<userID for CIM server>)  
GROUP(<group CIM server userID belongs too>))  
  
SETOPTS RACLIST(STARTED) REFRESH
```

Customizing the USS shell for CIM server startup

To start the CIM server from a Unix System Services shell, you must set the required environment variables which are contained in file */usr/lpp/wbem/install/profile.add*. You can add the contents of the profile to */etc/profile* to enable the CIM server environment for all users of the Unix System Services shell or to the individual *~/.profile* of each user. The environment variable PEGASUS_HOME should reflect your environment, that is, point to the installation directory of the CIM Runtime environment.

Customizing environment variables

Environment variables are set in file *cimserver.env* if the CIM server will run as started task, or, if you use the CIM server from the USS command prompt, they are set in USS *.profile* in the home path of the userID from which the CIM server is started.

Setting of the trace variables is not required for normal operation.

The environment variable file *cimserver.env* can be found in the HFS at */etc/wbem/cimserver.env*. After installation, you can still find the originally shipped version in */usr/lpp/wbem/*. Set the following environment variables contained in this file to start the CIM server:

PEGASUS_HOME

Must be set to the HFS directory where the CIM server is installed. By default this is */usr/lpp/wbem*.

LIBPATH

Must include the CIM server's lib and provider HFS directory paths. By default this is set to *lib:provider*.

OSBASE_TRACE

Defines the trace level for the z/OS CIM instrumentation. Valid values range from 0 through 4, where 4 provides the most details.

OSBASE_TRACE_FILE

Defines the filename for the z/OS CIM instrumentation traces.

PATH Only for running the CIM server or any of the CIM server command-line utilities in Unix System Services. Must include the CIM server's bin HFS directory path so that the CIM server's executable programs are

automatically found when you enter the according command at the Unix System Services command prompt. By default this is set to */usr/lpp/wbem/bin*.

The following variables starting with `RMF_` only apply when RMF is installed and you use the `RMF @server` monitoring providers:

RMF_CIM_HOST

defines the target TCP/IP address or hostname of the z/OS MVS image on which the DDS responsible for this system is running.

RMF_CIM_PORT

defines the TCP/IP port number of the DDS (default: 8803).

RMF_CIM_TRACE

defines the trace level of the RMF CIM provider. Valid values range from 0 through 4, with 0 providing no trace and 4 providing all information possible.

RMF_CIM_TRACE_FILE

defines the file name for storing the trace data for the z/OS RMF CIM instrumentation.

RMF_CIM_BENCH

used for performance benchmarks, for example, to identify the response time of the underlying RMF infrastructure. If this variable is set to 1, the RMF CIM provider will print some benchmarking information on various RMF operations, suitable for RMF development.

Setting up security for the CIM server

The security design of the CIM server on z/OS is based on the open source implementation, but has been extended to comply with z/OS security.

The z/OS implementation of the CIM server requires each user to have a real z/OS user ID. Only users who have been successfully authenticated with the z/OS security product and who have been granted access to the CIM server will be able to execute requests through the CIM server. This chapter provides an overview of the CIM server's security features and describes the details on how to set up these features.

Overview of CIM server runtime environment security

Note:

This document describes how to set up security using RACF as security product. However, you can use any other suitable security product for this purpose.

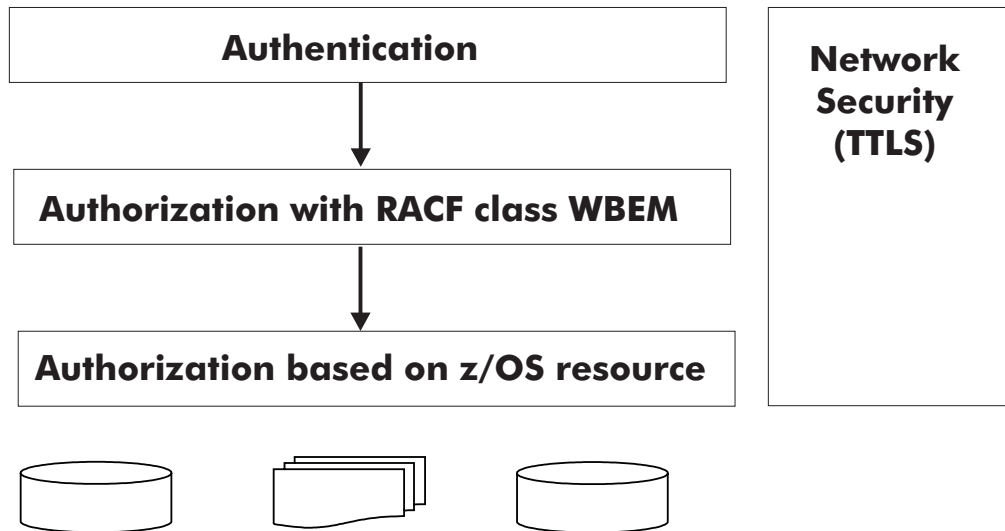


Figure 4. Authentication and authorization for the CIM server

The CIM server runtime environment security is split into authentication and authorization. Authentication is always enabled for the CIM server. The CIM server supports authorization via the RACF class WBEM, in which currently the single profile CIMSERV restricts access to the CIM server.

For authorization purposes to specific z/OS system resources, the CIM server executes requests under the user ID from which the request was generated. To do this, the CIM server uses thread-level security, which is provided by the Unix System Services. Additionally, the CIM server is enabled for the Enhanced Security model. Under the Enhanced Security model, the CIM server does not load any dynamic link library that is not program controlled, in particular it does not load any such provider DLL.

Setting up security for the CIM server includes the following steps:

1. Define a RACF class and profile for the CIM server.
2. Define a userID for the CIM server and grant it access to the CIM server's RACF profile.
3. Configure the CIM server's authorization model.
4. Grant client users and administrators access to the CIM server.
5. Optionally configure https for the CIM server.

The following chapters describe how each step of security can be enabled and configured.

Step 1: Defining a RACF class and profile for the CIM server

Access to the CIM server is controlled through RACF class WBEM. How to define a new class in RACF through the dynamic CDT feature of the *z/OS Security Server* is described in the following.

To be able to build the dynamic class WBEM, you must first build the class descriptor table (CDT) using the following RACF command:

```
SETROPTS CLASSACT(CDT) RACLIST(CDT)
```

By adding a profile to the IBM class named CDT, you can create a new class definition. This profile then represents a dynamic class. The segment CDTINFO is used to define the class attributes. Definition of dynamic class WBEM can be done with the following RACF commands:

```
RDEFINE CDT WBEM UACC(NONE)
CDTINFO(CASE(UPPER)
        FIRST(ALPHA)
        OTHER(ALPHA,NUMERIC)
        MAXLENGTH(246)
        MAXLENX(246)
        KEYQUALIFIERS(0)
        PROFILESALLOWED(YES)
        POSIT(200)
        RACLIST(REQUIRED)
)

SETROPTS CLASSACT(CDT) REFRESH

SETROPTS RACLIST(CDT) REFRESH
```

The default values shown above (except POSIT(200)) are expected by the CIM server; do not use different values as this can yield unpredictable results. For a more detailed description of how to create a new class within RACF dynamic CDT, please see *z/OS Security Server RACF Security Administrator's Guide*.

To activate the new class issue:

```
SETROPTS CLASSACT(WBEM) RACLIST(WBEM)
```

After creating and activating the WBEM class, you must create the CIMSERV profile within this class through which users then can be granted access to the CIM server.

The following example illustrates the flow of RACF commands that are required to define a profile named CIMSERV in this class:

```
SETROPTS CLASSACT(WBEM)

SETROPTS RACLIST(WBEM)

RDEFINE WBEM CIMSERV

SETROPTS CLASSACT(WBEM) RACLIST(WBEM) REFRESH
```

Step 2: Define a userID that runs the CIM server

Either select an existing userID or create a new userID that runs the CIM server. Depending on the security model under which the CIM server runs (see “Step 3: Configuring the CIM server’s authorization model” on page 12), a privileged user (UID=0) may be required.

In addition, the CIM server’s userID must be given CONTROL access to profile CIMSERV in class WBEM. The following example shows the required RACF commands to achieve this, where the userID PEGADM was chosen for the CIM server:

```
PERMIT CIMSERV CL(WBEM) ACCESS(CONTROL) ID(PEGADM)

SETROPTS CLASSACT(WBEM) RACLIST(WBEM) REFRESH
```

When running the CIM server as a started task, this userID also needs to be associated with the started task procedure CFZCIM as described in “Customizing started task procedure CFZCIM” on page 7.

Step 3: Configuring the CIM server's authorization model

The CIM server can be run with two different authorization models, depending on whether the profile BPX.SERVER is defined in the FACILITY class or not. In any case, the CIM server follows a resource-based authorization model, which means that user requests are run in separate threads, for which the security context is switched to the userID of the requestor. So any z/OS system resource that is accessed directly through a CIM provider executing a user requests in such a thread, will be accessed under the requestor's userID and thus, authorization checks occur against this userID.

These checks will occur in addition to the general access check for the CIM server through the CIMSERV profile in class WBEM.

Note: The definition of BPX.SERVER is not specific for the CIM server, but has system wide implications for all programs running on the z/OS system.

To let the resource based authorization security work properly, setup of one of the following conditions is required:

1. If no profile BPX.SERVER exists in the FACILITY class:
 - The CIM server runtime environment must run as privileged user. (The userID under which the CIM server runs must be defined with UID=0).
2. With BPX.SERVER (Enhanced Security model) enabled, ensure the following conditions:
 - The CIM server runtime environment user ID must have read or update access to BPX.SERVER.
 - All CIM server runtime environment libraries within HFS, as well as all installed providers, are flagged as program controlled. By default, all libraries shipped with the CIM server are flagged as program controlled. If additional provider libraries are installed, it may be required to set the program control flag manually using the extattr +p <libname> command.
 - **Note:** Language Environment modules SCEERUN and SCEERUN2 need to be program controlled.
 - The CIM server runtime environment runs in its own address space. To run the CIM server runtime environment in its own address space, either start it using the provided started task procedure or set environment variable `_BPX_SHAREAS=NO` in your z/OS Unix System Services shell before starting it with the *cimserver* command.

Note the difference in authorization check by access type, performed by the CIM server runtime environment userID, to BPX.SERVER:

- When permitted to BPX.SERVER with UPDATE access, z/OS resource access authority is based on the requesting client's authority.
- When permitted to BPX.SERVER with READ access, z/OS resource access authority is based on the server program's **and** the requesting client's authority. For example, even if the client user has a wider resource access authority than the server, the result is the least common denominator type of access authority.

If any of these requirements is not met, the CIM server will not start, but issue an according error message in the logs.

The CIM server runtime environment checks the program control flag while running under Enhanced Security (BPX.SERVER) before loading any dynamic libraries in order to not get its address space marked dirty and to make sure no

code is executed that has not been audited by a system programmer. Therefore all provider libraries must have set the program control flag, too.

For a more detailed description of how to define enhanced security for the CIM server, see Chapter 18 - Preparing security for servers in *z/OS UNIX System Services Planning*.

Step 4: Granting clients access to the CIM server

The CIM server authenticates users with the *z/OS Security Server* to determine which users can log into it. Authentication is performed for every new connection (local or remote) before a user is granted access to the CIM server.

For the CIM server for z/OS, users log in over HTTP or HTTPS using basic authentication. When logging in, users are authenticated using their z/OS user ID and password as defined, for example, in the Resource Access Control Facility (RACF®).

A CIM user must have at least READ access to the RACF profile CIMSERV in class WBEM to access the CIM server. In order to use any of the administrative command-line tools of the CIM server, as described in “CIM server command-line utilities” on page 18, a user instead requires CONTROL access to the CIMSERV profile.

The following example shows how to define READ access for a client userID called PEGUSR:

```
PERMIT CIMSERV CL(WBEM) ACCESS(READ) ID(PEGUSR)
```

```
SETOPTS CLASSACT(WBEM) RACLIST(WBEM) REFRESH
```

In addition, the CIM server's userID must be defined as a surrogate of the client userID. This is required to enable the CIM server to switch the security context to the client's userID for the thread(s) in which a user's request is executed. The following sample shows the according RACF commands, where the client userID is PEGUSR and the CIM server's userID is PEGADM:

```
RDEFINE SURROGAT BPX.SRV.PEGUSR UACC(NONE)
```

```
PERMIT BPX.SRV.PEGUSR CL(SURROGAT) ACCESS(READ) ID(PEGADM)
```

```
SETOPTS GENERIC(SURROGAT) REFRESH
```

Step 5: Configuring the CIM server HTTPS connection using TTLS

The CIM server runtime environment can profit from the new TTLS functionality introduced with z/OS V1.7. For a more detailed explanation about TTLS please see *z/OS Communications Server: IP Configuration Guide*.

To configure the CIM server to use the IP port for HTTPS, the configuration property *enableHttpsConnection* should be set to true. The default port for HTTPS communication is 5989 and it is not recommend to change the default.

Note: It is important to understand that only enabling HTTPS for the CIM server will not be sufficient to configure SSL/TLS encryption for the communication. You also must configure TTLS for the CIM server's HTTPS port.

The easiest way to setup TTLS for the CIM server is to create a rule that restricts TTLS to the CIM server port 5989 and to inbound TCP/IP communication:

```
TTLRule                      PegSrvRuleR
{
  RemotePortRange            5989
  Direction                  Inbound
  TTLSGroupActionRef         grp_act1
  TTLSEnvironmentActionRef   PegSrvEnvironment
}
```

For the client the same applies, just for outbound TCP/IP communication:

```
TTLRule                      PegClRuleR
{
  RemotePortRange            5989
  Direction                  Outbound
  TTLSGroupActionRef         grp_act1
  TTLSEnvironmentActionRef   PegClEnvironment
}
```

If you set up the CIM server HTTPS connection this way, the CIM server configuration does not influence other parts of the systems.

Configuring the CIM server

The CIM server provides the ability to set a number of configuration options. You must start the CIM server before you can configure it.

Many tasks and operations for the CIM server are performed under z/OS® UNIX® System Services, ideally within a telnet session.

Starting and stopping the CIM server

Start the CIM server from either the USS command prompt or as a started task, as described in the following sections.

Running the CIM server from the USS command prompt

To start the CIM server from the USS shell, type the `cimserver` command at the command prompt of a z/OS UNIX System Services shell. To run the CIM server runtime environment in its own address space, set the environment variable `BPX_SHAREAS=NO` in your z/OS Unix System Services shell beforehand.

Stop the CIM server by entering `cimserver -s` at the command line.

To let the *cimserver* command work from a USS shell, make sure you have completed the configuration steps described in “Customizing the USS shell for CIM server startup” on page 8.

Running the CIM server as started task

The standard way to start the CIM server on z/OS is through the started task CFZCIM. Thus you can start the CIM server from the z/OS console by entering `S(TART) CFZCIM`

Before starting the CIM server for the first time through CFZCIM, please customize the procedure to match the correct installation paths in the DD statements and also update the *cimserver.env* file installed in */etc/wbem* to match the correct installation paths for the CIM server. When using the default installation directory

/usr/lpp/wbem, no changes to either CFZCIM nor *cimserver.env* are required. For information on how to customize CFZCIM and *cimserver.env* refer to “Customizing the CIM server” on page 7.

When the CIM server was started through CFZCIM, it can also be stopped from the console by entering

```
(STO)P CFZCIM
```

Please note that the MODIFY command is currently not supported for the CIM server.

After a successful start of the CIM server, the following message is shown on the console and stored in the CIM server standard log file:

```
CFZ00001I: PGS10030: Started CIM server version 2.4.1 for z/OS.
```

CIM server configuration

The default configuration setting for the CIM server should work for the majority of environments. However, the CIM server provides command-line utilities to change the default configuration settings (see “CIM server command-line utilities” on page 18).

Changing current configuration properties

Use the *cimconfig* command to dynamically change the current configuration properties of the CIM server. Typically, the only current configuration properties that you can change are the *shutdownTimeout* property and the logging and tracing properties. You can update the current configuration while the CIM server is running.

After a restart of the CIM server, these changes will be reset to the planned configuration values. For making permanent changes, you must change the planned configuration values.

Changing planned configuration properties

You can use the *cimconfig* command with the *-p* option to change values for the planned configuration properties, irrespective of whether the CIM server is running or stopped. If you change the planned configuration properties, those changes do not take effect until the CIM server is restarted. Then the planned configuration properties become the current configuration properties.

CIM server configuration properties

You can change the following configuration properties with the *cimconfig* command:

Table 2. CIM server startup options

Option name	Description	Default value	dynamic Y/N
daemon	The foreground/background process property. Set daemon to false to run the CIM server as foreground process.	true For running the CIM server as a started task, this option is set to false.	N

Table 2. CIM server startup options (continued)

Option name	Description	Default value	dynamic Y/N
enableAssociationTraversal	The CIM server traverses associations along the inheritance tree. Disabling this increases performance, but some associations become not resolvable.	true	N
enableHttpConnection	The HTTP connection to the CIM server. Enables and disables connections to the CIM server over HTTP. When turned off only local connections are accepted.	true	N
enableHttpsConnection	The HTTPS connection to the CIM server. Enables and disables secure connections to the CIM server via HTTPS. Note that it is not sufficient to turn on this option, but you must also enable an SSL/TLS connection through the TTLS feature at the z/OS Communications Server as described in “Step 5: Configuring the CIM server HTTPS connection using TTLS” on page 13.	false	N
enableProviderProcesses	This option is currently not supported for z/OS.	false	N
enableRemotePrivilegedUserAccess	The remote privilege for users. Enables and disables access for users with UID 0.	false	N
httpAuthType	The authentication type. Currently only Basic authentication is supported (see also “Setting up security for the CIM server” on page 9).	Basic	N
httpPort	The port to listen for HTTP requests. It is not recommended to change this value.	5998	N
httpsPort	The port to listen for HTTPS requests. TTLS should be configured to use this port. It is not recommended to change this value.	5989	N
logdir	The name of the directory to which the log files are written.	/var/wbem/logs	Y
logLevel	The detail level for logging. Possible values are TRACE, INFORMATION, WARNING, ERROR or FATAL. TRACE returns the most detailed log information.	INFORMATION	Y
messageDir	The message bundle directory. Do not change the default.	msg	N

Table 2. CIM server startup options (continued)

Option name	Description	Default value	dynamic Y/N
providerDir	The name of the directory where the provider libraries reside. You can specify multiple directories here, separated by a colon (':'). Since the CIM server has its own set of providers, its lib directory always needs to be present in the list of provider directories. When adding new provider directories, it is also recommended to update the LIBPATH environment variable according to the new values of providerDir. This is required, because a provider may need other supplemental DLLs, which the CIM server is not aware of and therefore would otherwise fail to load. Note: You must provide the full path for all directories when changing the default.	lib:provider	N
repositoryDir	The name of the directory for the repository.	/var/wbem/repository	N
repositoryIsDefaultInstanceProvider	The CIM server repository serves as the default provider for CIM instances when no dynamic provider has been registered for a CIM class.	true	N
shutdownTimeout	The timeout value in seconds that the CIM server uses to wait for the shutdown process to complete. This value includes terminating active providers.	10 seconds	Y
slp	The CIM server uses the SLP Protocol to announce itself over the network.	true	N
tempLocalAuthDir	The directory used for temporary file storing to secure local authentication.	/tmp	N
traceComponents	Use this parameter to specify the component(s) that you want to trace. In combination with parameter <i>traceLevel</i>, this option turns on and off tracing for single components of the CIM server. Among the components that you can specify with this parameter are the following: • Channel • XmlWriter • CimData • ProvManager • Authorization • Authentication • Thread Setting this value to all will enable tracing for all components. If the <i>traceComponents</i> parameter is not set to any component, tracing is off regardless of the <i>traceLevel</i> parameter setting.		Y

Table 2. CIM server startup options (continued)

Option name	Description	Default value	dynamic Y/N
traceFilePath	The name of the directory where the trace file is written.	/var/wbem/cimserver.trc	Y
traceLevel	The trace level. Turns tracing on/off and controls the level of detail for tracing. Tracing should only be used for debugging purposes and is, by default, disabled. You can, however, enable the tracing mechanism by specifying the trace level and the components that you want to be traced with this configuration parameter. Choose from one of the following trace levels: level 1 Function entry and exit level 2 Basic flow, trace messages, low data detail level 3 Interfunction logic flow, medium data detail level 4 All information, high data detail Note: This does not include tracing for the providers.	1	Y

CIM server command-line utilities

The CIM server includes a set of command-line utilities that you can use to control or change the CIM server environment. During normal use, you should rarely need to use these commands.

You must run all of the command-line utilities from a z/OS UNIX System Services shell. The command-line utilities need the same environment setup as described in “Configuring the CIM server” on page 14. Users of these command-line utilities must have CONTROL access to profile CIMSERV in class WBEM.

The commands include:

cimmof/cimmofl

These commands are used to compile provider registrations and to compile Common Information Model (CIM) class descriptions written in the managed object format (MOF) language. The compiled information is put into the class schema stored in the repository.

cimmofl is a version of **cimmof** that does not use the CIM server. The CIM server must be stopped before using this command (see “cimmof/cimmofl” on page 19).

cimconfig

This command configures the startup options for the CIM server. Depending on the property being configured, the CIM server may need to be restarted after using this command (see “cimconfig” on page 20).

cimprovider

This command enables or disables a registered provider, primarily during testing. The CIM server must be running to use this command (see “cimprovider” on page 22).

You can specify most options provided by the utilities in two ways:

- a short form introduced by a single dash, for example -f<file>
- a long form introduced by a double dash, for example --file=<file>

cimmof/cimmofl

Description

These commands are used to compile provider registrations and to compile CIM class descriptions written in the MOF language and store the information in the repository. For **cimmof**, the CIM server must be started before using this command. **cimmofl** runs without using the CIM server. This version of the MOF compiler does only limited error checking, may handle instance operations incorrectly, and does not protect against concurrent access to the CIM repository. Therefore, **cimmof** is the recommended MOF compiler.

The CIM server MOF compiler is a command-line utility that compiles MOF files (using the MOF format defined by the DMTF CIM Specification) into a CIM server repository. It allows compiling from structures of MOF files using the include #pragma and can either compile into a CIM server repository or check the syntax on the MOF files. The compiler requires that the input MOF files are in the current directory or that a fully qualified path is given. MOF files that are included using the #pragma include must be in the current directory or in a directory specified by a -I command-line switch.

Syntax

```
cimmof [-h] [-E] [-w] [-uc] [-aE | -aV | -aEV] [-R repository]
[-I path] [-n namespace] [--xml] [--trace | -trace=<tr_file>] [mof_file...]

cimmofl [-h] [-E] [-w] [-uc] [-aE | -aV | -aEV] [-R repository]
[-I path] [-n namespace] [-xml] [-trace | -trace=<tr_file>] -ffile
```

Options

-h, - help

Prints out usage message with command definitions.

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

-w Suppresses warning messages.

-uc Allows 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. 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>, --CIMRepository=<path>

If specified, this overrides the current repository path used by the CIM server. Specify an absolute or relative path.

-I<path>

Used to specify a path to the included MOF files.

mof_file

Used to specify one MOF file or a list of MOF files to compile.

-f<file>, --file=<file>

Used to specify a file that contains a list of MOF files to compile.

-n<path>, --namespace=<path>

Used to override the 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, --trace=<tr_file>

Used to write trace information to a file. If tr_file is omitted, the output destination is standard output. Those files are written with ASCII encoding.

Example

cimmofl -w -Rtestrepository -I./myDir myDir/CIM_Schema28.mof

In this example, the managed object format (MOF) file that is located in directory myDir with the name CIM_Schema28.mof is compiled. CIM_Schema28.mof includes #pragmas for other MOF files that are also in the myDir directory. Therefore an include (-I) option is required for the myDir 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.

For using the **cimmof/cimmofl** commands against the CIM server namespaces (root# PG_Internal, root# PG_InterOp), a user needs to have CONTROL access to profile CIMSERV in class WBEM.

cimconfig

Description

Use the **cimconfig** command to manage CIM server configuration properties. You can get, set, unset, or list these properties. See “Configuring the CIM server” on page 14 for more information.

Syntax

- **cimconfig -g name [-c | -p | -d]**
- **cimconfig -s name=value [-c | -p]**
- **cimconfig -u name [-c | -p]**
- **cimconfig -l [-c | -p]**

The first form of this command provides the current, planned, and default value of the specified configuration property (get).

The second form lets you set the current value and planned value of the specified configuration property to the specified value (set).

The third form lets you set the current and planned value of the specified configuration property to its default value (unset).

The last form of this command lists all configuration properties (list). Specifying the -c or -p options provides a listing of all current or planned configuration property names and values.

Specifying no options with the **cimconfig** command shows the usage message.

Options

The **cimconfig** command recognizes the following options:

-g name [-c]

Gets the current value of the specified configuration property. Returns an error when the CIM server 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 CIM server is not running.

-s name=value [-c]

Indicates that a configuration property is to be added or updated by setting it to the specified value. Returns an error when the CIM server 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 [-c]

Indicates that the current value of the specified configuration property is to be reset to the default value. Returns an error when the CIM server 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 configuration properties. Returns an error when the CIM server is not running.

-l -c

Displays the name and value pair of all current configuration properties. Returns an error when the CIM server is not running.

-l -p

Displays the name and value pair of all planned configuration properties.

Note: You can use the **cimconfig** command to set the current or planned configuration properties of the CIM server. 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 do not take effect until the CIM server is restarted. When the CIM server is started, the planned configuration properties become the current configuration properties.

For using the **cimconfig** command, a user needs to have CONTROL access to profile CIMSERV in class WBEM.

cimprovider

Description

The **cimprovider** command lets you disable, enable, unregister, and list registered CIM providers or CIM provider modules and module status. If a CIM provider is disabled, the CIM server rejects any requests to the provider. If a CIM provider is enabled, the CIM server forwards requests to the provider. If a CIM provider is unregistered, the CIM server will no longer have any information about the provider. In order to use the **cimprovider** command, the CIM server must be running and the specified provider module (a grouping of providers in the same load library (dll)) or provider must be registered with the CIM server.

Syntax

- **cimprovider -d -m module**
- **cimprovider -e -m module**
- **cimprovider -r -m module [-p provider]**
- **cimprovider -l [-s | -m module]**

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 are 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 a new request.

The third form of **cimprovider** removes (unregisters) the specified provider module and all of its contained providers or a single specified provider in the specified provider module. After removal, 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 registered provider modules and module status or all providers in the specified provider module. To list all providers in all modules, type a **cimprovider -l** command, followed by **cimprovider -l -m** for each listed module.

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

Options

The **cimprovider** command recognizes the following options:

- | | |
|-----------|---|
| -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 registered provider modules.
-m module	Specifies the provider module for the operation.
-p provider	Specifies the provider for the operation.
-s	Displays the status of provider modules.

Limitations

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

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 an 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 (un-register) the MyProvider provider 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, which are in the myProviderModule provider module.

For using the **cimprovider** command, a user needs to have CONTROL access to profile CIMSERV in class WBEM.

Supported CIM operations

While the z/OS CIM server supports all of the CIM operations from the DMTF's *CIM Operations over HTTP* specification, only a specific subset of operations is supported by the OS management CIM providers delivered with this release of z/OS.

The following operations are available for all OS management classes or for association classes.

Available for all OS management classes:

- EnumerateInstanceNames
- EnumerateInstances
- GetInstance

Additionally available for all association classes:

- Associators
- AssociatorNames
- References
- ReferenceNames

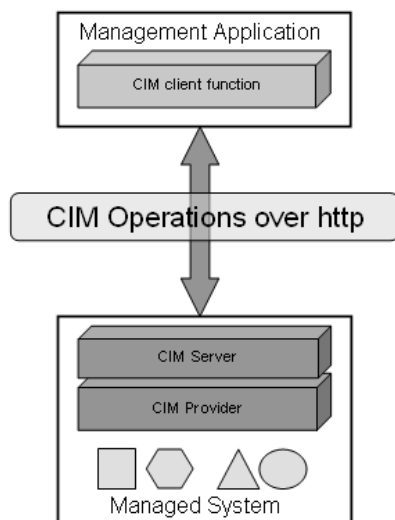


Figure 5. How CIM operations interact between CIM clients and the CIM server

Sample client applications

An OCO sample client application **pszos** is delivered as executable with the product in file `/usr/lpp/wbem/bin/pszos`. You can use it as an installation verification program. It displays information about the z/OS system and lists currently active USS processes.

You invoke this program as job CFZIVP contained in SYS1.PROCLIB or from the USS command line as `pszos 127.0.0.1`.

On successful completion, it generates an output similar to the following:

```

pszos Main started ...
pszos::_connectToServer(127.0.0.1:5988) called
pszos::_connectToServer - success
> Found Computer System : BOEMT11.boeb.de.ibm.com
                        (CPUID: 026F7A2084, VMGuestID: MT11)
> Found Operating System : MT11 (Version: 01.07.00, Sysplex: MT1PLEX,
                        FreeMem: 30192)
> Found the following USS processes:
--> 1 - BPXPINPR
--> 5 - EZBTCPIP
--> 7 - EZBTSSL
--> 16777227 - inetd
--> 67108877 - EUVFSKDC
--> 67108883 - otelnetd
--> 50331670 - cimserver
--> 83886103 - PAGENT
--> 27 - pszos
  
```

The source code for a sample client monitoring application is printed in the *z/OS RMF Programmer's Guide*.

Troubleshooting

This chapter contains the following subsections:

- “Problem determination”
- “Error messages”

You can find further helpful information in:

- “Logging” on page 45
- “Checking the CIM server logs” on page 45
- “CIM server messages” on page 45

Problem determination

If you are having trouble while using the CIM server you can trace the CIM server activities. Therefore, use the following startup options for the *cimconfig* command (see “CIM server configuration properties” on page 15):

- `traceLevel`
- `traceComponents`

These can be set dynamically:

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

or permanently:

```
cimconfig -p -s traceLevel=2
```

or temporarily on the USS command line of the CIM server:

```
cimserver traceLevel=2 traceComponents=all
```

Error messages

Error: ICH14080I

Warning: RACF detected a possible error in the dynamic class descriptor table, entry WBEM, error code 01. The class is available for further processing. The class name does not contain a national character nor a number. To assure IBM does not create an IBM-defined class in the future by this same name, you should choose a class name which contains at least one national character or a number.

When or where seen: RACF setup of dynamic class WBEM

Solution: Ignore the warning.

Error: CIM_ERR_ACCESS_DENIED

Access to a CIM resource was not available to the client: "Not authorized to run <name of a CIM Operation> in the namespace root/PG_Internal"

When or where seen: Client application / Details in the CIM server trace log

Solution: Permit the user ID to execute a configuration command with CONTROL access to Security profile CIMSERV in CLASS WBEM.

Error: CIM_ERR_ACCESS_DENIED

Access to a CIM resource was not available to the client: "Not authorized to run <name of a CIM Operation> in the namespace root/PG_InterOp"

When or where seen: Client application / Details in the CIM server trace log

Solution: Permit the user ID to execute a configuration command with CONTROL access to Security profile CIMSERV in CLASS WBEM.

Error: CIM runtime environment user ID requires either READ access to BPX.SERVER or it must be user ID 0.

When or where seen: The CIM server error log after CIM server fails to start

Solution: Either permit the user ID READ access to BPX.SERVER if BPX.SERVER is set up, or run the command under a privileged user ID (UID 0).

Error: CIM runtime environment user ID requires CONTROL access to profile CIMSERV in class WBEM.

When or where seen: The CIM server error log after CIM server fails to start

Solution: The CIM server startup fails because the CIM server user ID fails to have CONTROL access to profile CIMSERV in class WBEM. Grant the CIM server user ID CONTROL access to profile CIMSERV in class WBEM.

Error: failed to bind to HTTP port

When or where seen: CIM server startup console message

Solution: The CIM server cannot start because it fails to listen on one of the ports 5988 (for http) or 5989 (for https). Either the CIM server is already running, another Server is listening on one of these ports, or the ports have been blocked in the TCPIP configuration.

Error: HTTP Error (401 Unauthorized)

When or where seen: Client application

Solution: The user authentication failed. The client application either did not provide user ID and password on a request at all, or the supplied user ID and password are not valid for the z/OS system on which the CIM server is running.

Permit the user ID to execute a client request with at least READ access to Security profile CIMSERV in CLASS WBEM. Check the server log for a detailed error report.

Client Side Error: CIM_ERR_ACCESS_DENIED

Access to a CIM resource was not available to the client: "EDC5139I Operation not permitted."

When or where seen: Client application / Details in the CIM server trace log

Solution: Permit the CIM server runtime environment user ID as surrogate for the requesting client user ID to use the command: PERMIT BPX.SRV.<client uid>
CL(SURROGAT) ID(<CIMServer UID>) ACCESS(READ)

Chapter 4. z/OS Management Instrumentation for CIM

The CIM standard provides the ability to develop management applications that work with systems management data. 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 DMTF web site.

IBM has developed providers for z/OS that support basic operating system information and some performance metrics. A CIM provider is the link between the CIM server and the system (see Figure 3 on page 5). This interface allows CIM to access and manage the resources. Each CIM provider makes accessible the resources it represents in a standard way.

Note: z/OS V1.7 does not implement all properties of the supported CIM classes described in this document. Those properties implemented by z/OS are documented in each of the following subchapters. For all CIM properties not implemented by z/OS, the CIM server returns no values.

The following CIM classes and associations are implemented as IBM-supplied providers to provide basic operating system information:

Base classes (page 30):

- IBMzOS_ComputerSystem: subclass of CIM_ComputerSystem
- IBMzOS_OperatingSystem: subclass of CIM_OperatingSystem
- IBMzOS_OSProcess: subclass of association CIM_OSProcess
- IBMzOS_Process: subclass of CIM_Process
- IBMzOS_RunningOS: subclass of association CIM_RunningOS
- IBMzOS_UnixProcess: subclass of CIM_UnixProcess

Processor classes (page 36):

- IBMzOS_CSProcessor: subclass of association CIM_SystemDevice
- IBMzOS_Processor: subclass of CIM_Processor

File System classes (page 38):

- IBMzOS_HostedFileSystem: subclass of association CIM_HostedFileSystem
- IBMzOS_NFS: subclass of CIM_NFS
- IBMzOS_UnixLocalFileSystem: subclass of CIM_UnixLocalFileSystem

Network classes ⁽¹⁾ (page 40):

- IBMzOS_EthernetPort: subclass of CIM_EthernetPort
- IBMzOS_CSNetworkPort: subclass of association CIM_SystemDevice
- IBMzOS_IPProtocolEndpoint: subclass of CIM_IPProtocolEndpoint
- IBMzOS_NetworkPortImplementsIPEndpoint: subclass of association CIM_PortImplementsEndpoint

Note (1): The z/OS Communications Server provides documentation of these CIM classes. For details refer to Considerations for Common Information Model (CIM) providers in *z/OS Communications Server: IP Configuration Guide*.

OS management Base classes

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

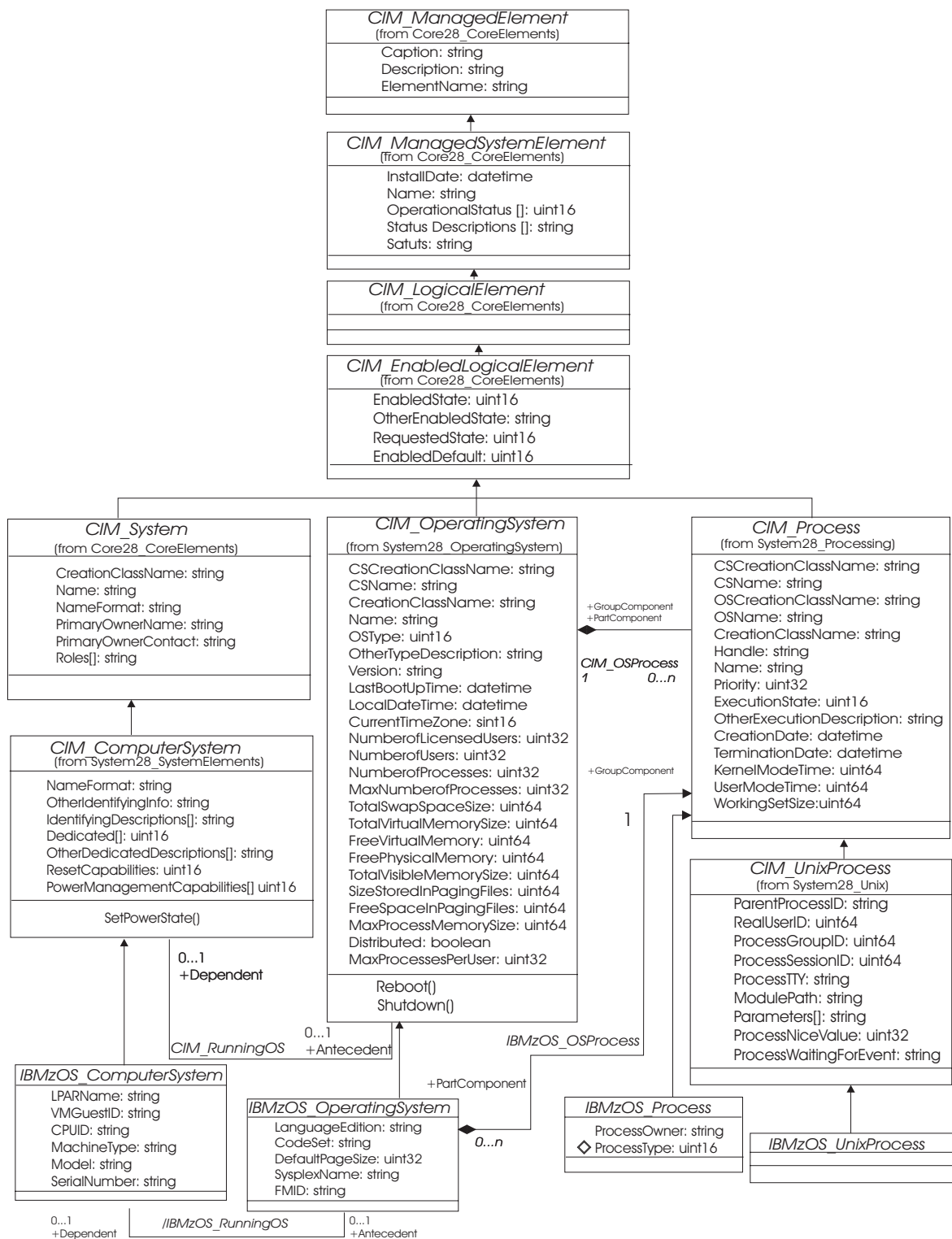


Figure 6. CIM classes extended by z/OS-specific classes

The MOF files that define these classes can be found in directory `schemas/os_management` relative to where the providers for z/OS have been installed. The default is `/usr/lpp/wbem/provider`.

CIM_ComputerSystem

This class represents either virtual or physical computer systems in the sense of a container inside which an operating system may run. This is the central class of the OS Management data model and aggregates all other resource classes.

The z/OS specific subclass is **IBMzOS_ComputerSystem**.

IBMzOS_ComputerSystem

This class provides basic computer system information such as computer name, and status information. A provider instruments this class so that it can be used by client applications to identify the managed system on which the provider is running (typically a server or an application).

The following properties are common for @server:

Property	Description
string Caption	Always set to <i>'IBM z/OS Computer System'</i> .
string Description	Always set to <i>'This is an IBMzOS_ComputerSystem'</i> .
string ElementName	Same as Name
string Name [key]	The fully qualified IP host name.
string CreationClassName [key]	Always set to <i>'IBMzOS_ComputerSystem'</i>
string NameFormat	Describes the format used to build the Name property. Always set to <i>'IP'</i> .
uint16 Dedicated[]	Indicates whether this is a special purpose system. Always set to 0 (not dedicated).

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

Property	Description
string LPARName	Name of the zSeries logical partition that makes up the computer system. If not running in LPAR mode, a blank string is returned here.
string VMGuestID	z/VM user ID of the virtual machine, of which the current z/OS image is a guest. If z/OS is not running as a guest under z/VM, a blank string is returned here.
string CPUID	Unique ID of the physical server on which this computer system is running.
string SerialNumber	IBM allocated number used to identify the server on which this computer system is running.
string MachineType	Brand name of the class for this z/OS server.
string Model	The name by which the server type is generally known.

CIM_OperatingSystem

This class represents a running operating system with its basic properties.

The z/OS specific subclass is **IBMzOS_OperatingSystem**.

IBMzOS_OperatingSystem

This class is for use by client applications to obtain basic properties of a running z/OS operating system.

The following properties are common for @server:

Property	Description
string Name [key]	The true name of the z/OS operating system.
uint16 OperationalStatus[]	Overall system status. Always OK for now.
uint16 OSType	Always 68 ("z/OS").
string Version	Version, release and modification of the operating system in the format of "VV.RR.MM". For example, for z/OS V1.7.0, this will return "01.07.00".
datetime LastBootUpTime	Time when the operating system was IPLed.
datetime LocalDateTime	Local time of the operating system
sint16 CurrentTimeZone	Time zone for the operating system, offset in minutes from GMT.
uint32 NumberOfUsers	The number of currently logged on TSO and Unix System Services users.
uint32 NumberOfProcesses	Total number of UNIX processes and active address spaces.
uint32 MaxNumberOfProcesses	The maximum number of processes configured in MaxProcSys.
uint64 MaxProcessMemorySize	The maximum number of KBytes of memory that can be allocated to a process (RLIMIT_AS).
uint64 TotalVirtualMemorySize	Total number of KBytes of virtual memory available to the operating system.
uint64 FreeVirtualMemory	Number of KBytes of virtual memory currently unused and available.
uint64 FreePhysicalMemory	Number of KBytes of physical memory currently unused and available.
uint64 TotalVisibleMemorySize	The total amount of physical memory (in KBytes) available to the operating system.
uint64 SizeStoredInPagingFiles	The total number of KBytes that can be stored in the operating system's page data sets.
uint64 FreeSpaceInPagingFiles	The total number of KBytes currently free in the operating system's page data sets.

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

Property	Description
string LanguageEdition	eServer specific extension for the language version of the OS. For z/OS always returns 'en-US'.
string CodeSet	eServer specific extension for the default OS code page. For z/OS this returns the code page for the CIM server process.
uint32 DefaultPageSize	eServer specific extension. The default size of pages used by the virtual memory management in units of bytes. Always 4096 for z/OS.
string SysplexName	The name of the z/OS Sysplex to which this operating system belongs.
string FMID	Function modification identifier of the z/OS operating system.

IBMzOS_OperatingSystem does not support the *Reboot()* and *Shutdown()* methods.

CIM_RunningOS

This class associates a computer system with the currently running operating system (see Figure 6 on page 30).

The z/OS specific subclass is **IBMzOS_RunningOS**.

IBMzOS_RunningOS

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.

CIM_Process

This class represents currently active processes on an operating system. For z/OS this is mapped to address spaces and USS processes.

The z/OS specific subclasses are:

- IBMzOS_Process (for address spaces)
- IBMzOS_UnixProcess (for USS processes)

IBMzOS_Process

This class provides basic process information such as process name, priority, and run-time state. Instances of class IBMzOS_Process are mapped to z/OS address spaces. Client applications can use this class to give clients an understanding of the processes (address spaces) running on the managed system within the context of their operating system.

The following properties are common for @server:

Property	Description
string Name	The name of the z/OS address space.
string Handle [Key]	The decimal representation of the address space ID(ASID).
uint32 Priority	The address space's dispatching priority.
datetime CreationDate	The time when the address space was created.
uint64 KernelModeTime	Not officially supported for z/OS.
uint64 UserModeTime	Not officially supported for z/OS.

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

Property	Description
string ProcessOwner	The primary z/OS user ID under which an address space was started.
uint16 ProcessType	The type of address space. Possible values are: 0 (Other), 1 (TSO User), 2 (Started Task), 3 (Job), 4 (System Address Space), 5 (Initiator).

Multiple inheritance

z/OS also provides the notion of a UNIX process through its UNIX System Services (USS) subsystem. In addition, those processes running under USS are supported by the extra IBMzOS_UnixProcess class which is derived from class CIM_UnixProcess. When a client enumerates all instances of class CIM_Process, it gets the complete list of z/OS address spaces, as well as all processes running under USS. However, if the client enumerates the instances of class IBMzOS_Process directly, it only gets the list of address spaces since class IBMzOS_UnixProcess is not derived from IBMzOS_Process but only from CIM_UnixProcess. Ideally, IBMzOS_UnixProcess should inherit from IBMzOS_Process, besides inheriting from CIM_UnixProcess, however, multiple inheritance is not the current standard in CIM version 2. For inheritance information of the mentioned classes refer to Figure 6 on page 30.

IBMzOS_UnixProcess

This class provides basic information about z/OS processes running in the Unix System Services (USS) subsystem. It supports all properties from CIM_Process plus a set of properties typical for Unix processes.

Class IBMzOS_UnixProcess is not derived from IBMzOS_Process, and therefore no instances of IBMzOS_UnixProcess are returned when a client enumerates the instances of class IBMzOS_Process, rather than class CIM_Process.

The following properties are common for @server:

Property	Description
string Name	The name of the z/OS Unix process. This is usually the name of the executable that started the process.

Property	Description
string Handle [Key]	The z/OS Unix process ID.
uint32 Priority	The process priority.
uint16 ExecutionState	The process state (ready, blocked, suspended, stopped, and so on).
datetime CreationDate	The time when the process was started.
uint64 KernelModeTime	Not officially supported on z/OS.
uint64 UserModeTime	Not officially supported on z/OS.
string ParentProcessID	The parent process ID.
uint64 RealUserID	The real user ID.
uint64 ProcessGroupID	The process group ID.
uint64 ProcessSessionID	The process session ID.
string ProcessTTY	The TTY currently associated with this process.
string ModulePath	The executing process's command path.
string Parameters[]	The operating system parameters provided to the executing process. These are the argv[] values.

Class IBMzOS_UnixProcess has no z/OS specific properties.

CIM_OSProcess

This class associates an operating system with the set of currently active address spaces and USS processes.

The z/OS specific subclass is **IBMzOS_OSProcess**.

IBMzOS_OSProcess

This class provides a link between the operating system and process(es) 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.

OS management Processor classes

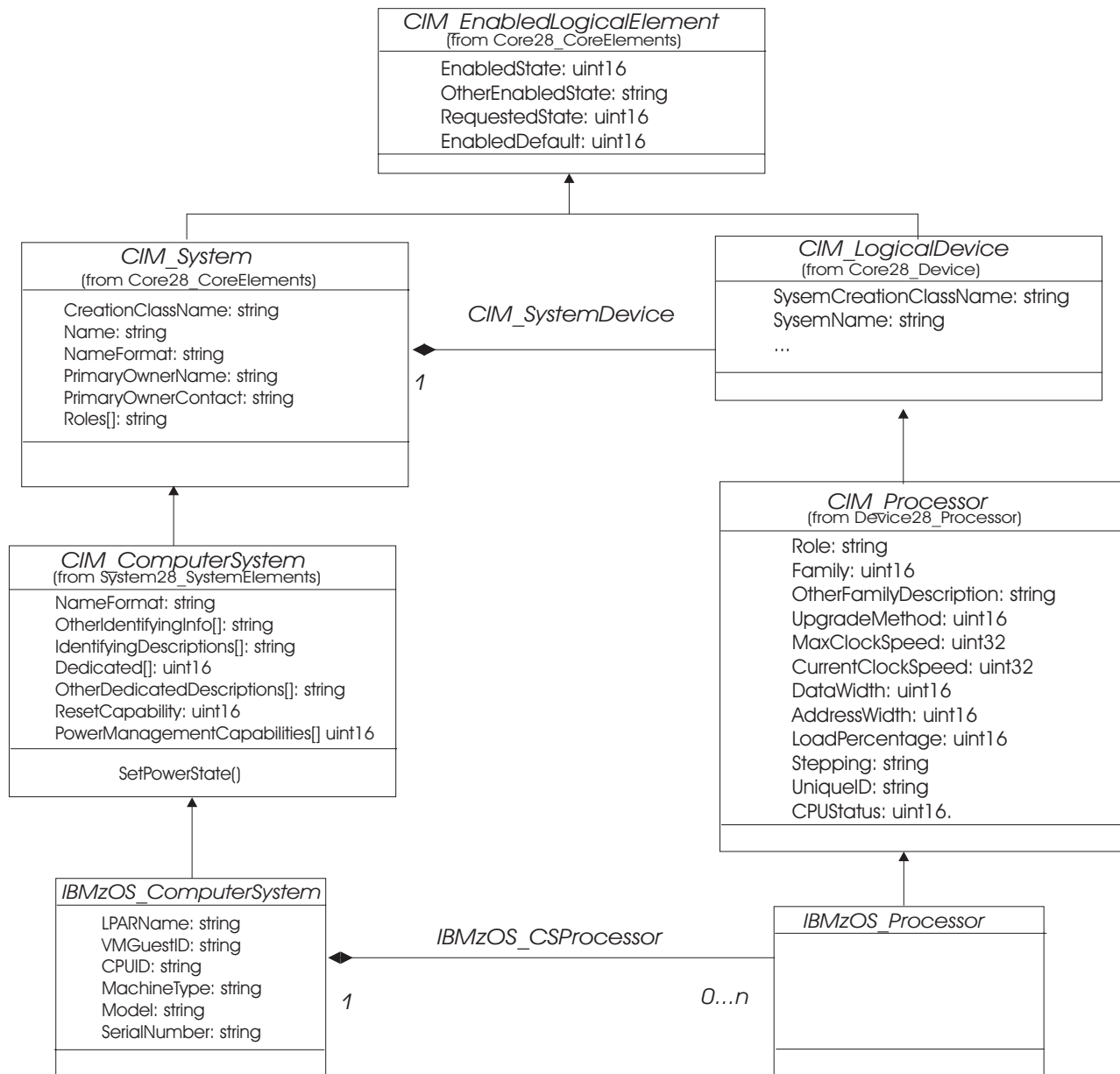


Figure 7. OS management Processor classes

CIM_SystemDevice association

This class associates a **ComputerSystem** with the instrumented processors.

The z/OS specific subclass is **IBMzOS_CSPProcessor**.

CIM_Processor

This class represents the physical processors that are available to the operating system.

The z/OS specific subclass is **IBMzOS_Processor**.

The following properties are common for @server:

Property	Description
string Caption	Always set to “zSeries logical processor”.
string Description	Always set to “This class represents instances of processors currently available to the z/OS operating system”.
string ElementName	Same as Name.
string DeviceID [key]	CPUID of the physical processor (PCCACPID).
string Role	Always set to “Central Processor”.
uint16 Family	200 (=“S/390 and zSeries Family”).
string OtherFamilyDescription	“S/390 and zSeries Family” or specific model like “z990”.
uint32 MaxClockSpeed	Not supported for z/OS.
uint32 CurrentClockSpeed	Not supported for z/OS.
uint16 LoadPercentage	For z/OS provided through RMF metrics provider only.
string Stepping	Not supported for z/OS.
string UniqueID	CPUID of the physical processor (PCCACPID).
uint16 CPUStatus	Not supported for z/OS.

IBMzOS_Processor

Class IBMzOS_Processor has no z/OS specific properties.

OS management File System classes

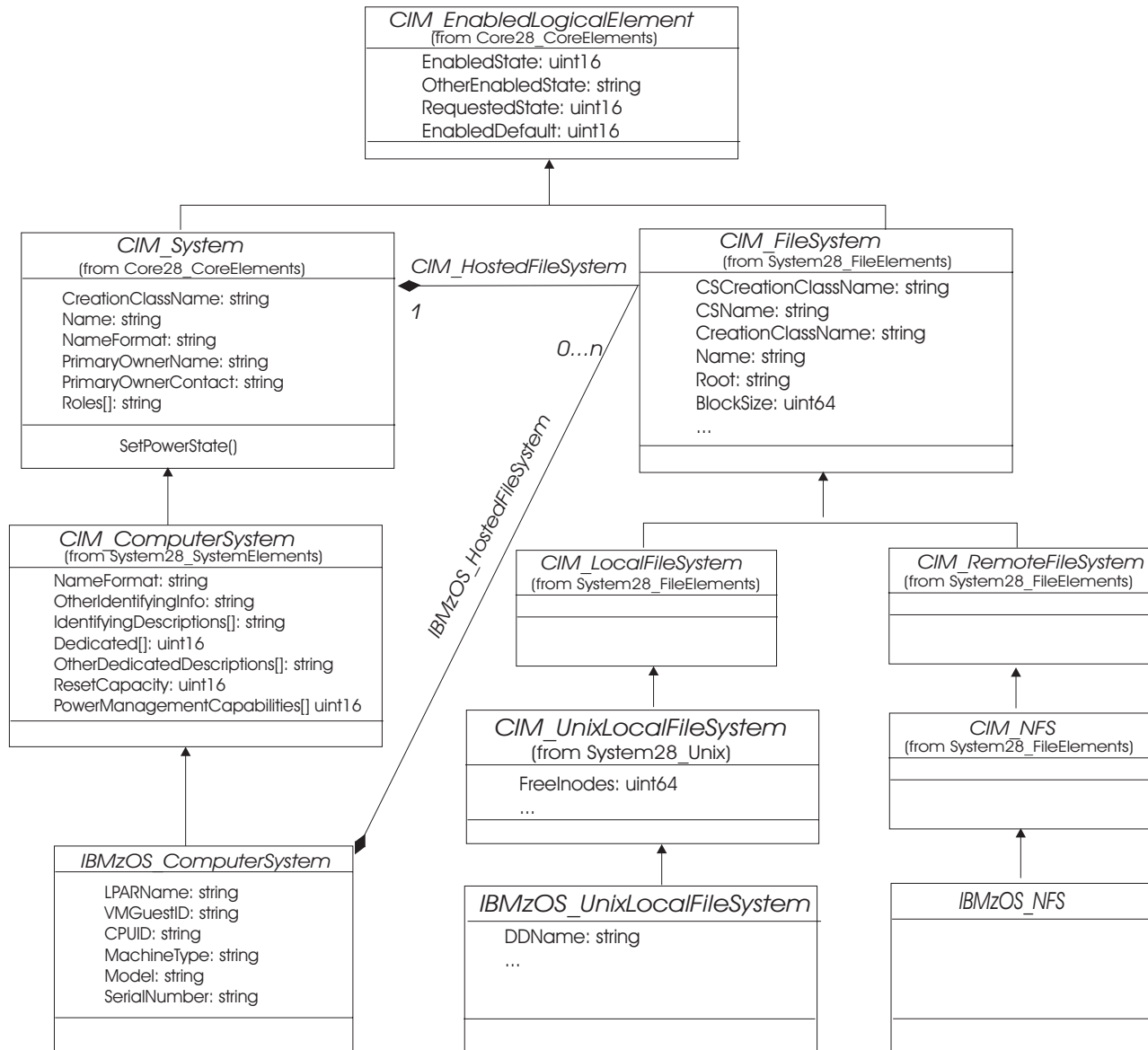


Figure 8. OS management FileSystem classes

CIM_HostedFileSystem

The **CIM_HostedFileSystem** association associates a **ComputerSystem** with the set of currently mounted USS file systems.

The z/OS specific subclass is **IBMzOS_HostedFileSystem**.

CIM_LocalFileSystem

This class represents file systems that are locally attached to a computer system. On z/OS, hierarchical file systems HFS and zFS are supported.

The z/OS specific subclass is **IBMzOS_UnixLocalFileSystem**.

The following properties are common for @server:

Property	Description
string Caption	Always set to 'z/OS hierarchical local file system'.
string Description	Always set to 'This class represents instances of currently mounted local hierarchical file systems'.
string ElementName	Same as Name.
string Name [key]	File system name (z/OS data set name).
string Root	Name of the directory where the file system is mounted.
uint64 FileSystemSize.	File system size in bytes.
uint64 AvailableSpace	Space available on the file system in bytes.
boolean ReadOnly	Indicates whether the file system is mounted read only.
string FileSystemType	File system type, for example "HFS".

IBMzOS_UnixLocalFileSystem

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

Property	Description
DDName	DD name that was specified on mount.
FSParentDeviceID	Device ID of the parent file system.
FSDeviceID	Device number which the STAT command will return for all files in this file system.
MountParameters	The parameters that were specified for the mount command.
FSOwner	MVS Owner ID of the file system.
FSTypeName	The file system type name from the PARMLIB statement.

CIM_RemoteFileSystem

This class represents file systems that are accessed remotely by a computer system. On z/OS, only NFS is supported.

The z/OS specific subclass is **IBMzOS_NFS**.

The following properties are common for @server:

Property	Description
string Caption	Always set to "z/OS mounted network file system".
string Description	Always set to "This class represents instances of currently mounted network file systems".
string ElementName	Same as Name.
string Name [key]	File system name (corresponds to the <i>file system</i> argument of the mount command).

Property	Description
string Root	Name of the directory where the file system is mounted.
uint64 FileSystemSize	File system size in bytes.
uint64 AvailableSpace	Space available the on file system in bytes.
boolean ReadOnly	Indicates whether the file system is mounted read only.
string FileSystemType	File system type, for example "HFS".

OS management Network classes

The classes described in this section are implemented by the z/OS Communication Server. For details on these CIM classes, refer to *z/OS Communications Server: IP Configuration Guide*.

Access to TCP/IP stack data is controlled by a security resource. Such a security resource is required if a user ID, associated with the client of the CIM server, is not defined as a z/OS UNIX superuser. The resource name is **EZB.CIMPROV.sysname.tcpname**. It is defined in the SERVAUTH class. Access is granted if the user ID associated with the client of the CIM server is permitted for READ access to the resource.

The providers are installed in the /usr/lpp/tcpip/lib HFS directory.

The z/OS CS CIM class definition and provider registration files are installed in the /usr/lpp/tcpip/mof HFS directory and are already integrated into the CIM server.

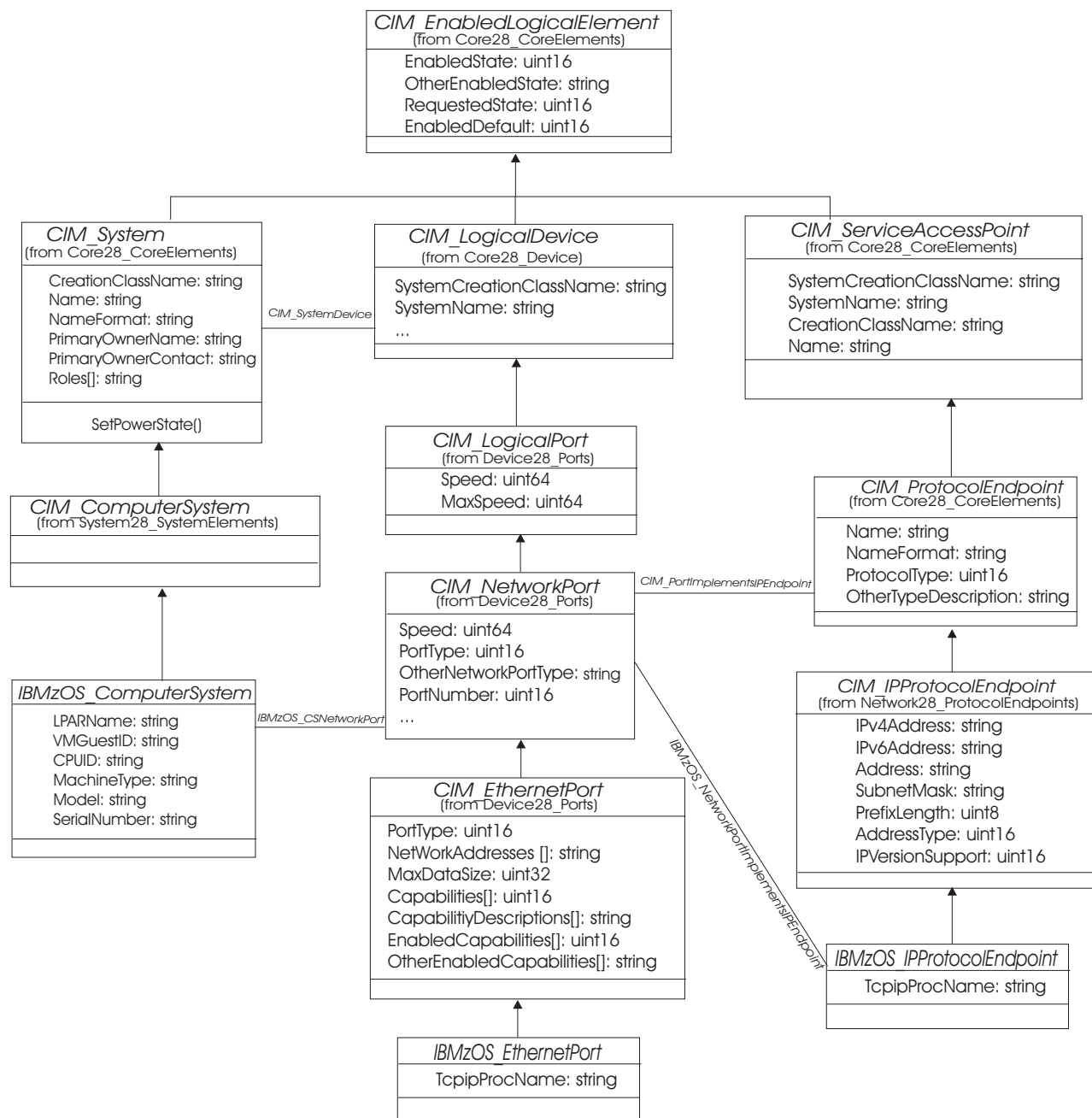


Figure 9. OS management Network classes

CIM_EthernetPort

This class represents network ports (interfaces) of type ethernet. For z/OS, all the ethernet interfaces configured to the TCP/IP stacks on the MVS image are supported.

The z/OS specific subclass is **IBMzOS_EthernetPort**.

The following properties are common for @server:

Property	Description
string Caption	Always set to <i>"IBMzOS EthernetPort"</i> .
string Description	Variable, depending on the type of interface, for example, <i>"IP Assist Queued Direct I/O Ethernet protocol port"</i> .
string ElementName	Same as Name.
string Name	The label by which the NetworkPort is known to the operating system ("tcpprocname_intfname").
uint16 EnabledState	Indicates whether the protocol endpoint is active or not.
string DeviceID [key]	Identifying information to uniquely name the ethernet port. ("tcpprocname_intfname").
uint64 Speed	The current bandwidth of the port in bits per second.
uint64 MaxSpeed	The maximum bandwidth of the port in bits per second. For z/OS, this is always the same value as Speed .
uint16 LinkTechnology	Always 2 (=Ethernet).
string OtherLinkTechnology	Not set for z/OS.

IBMzOS_EthernetPort

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

Property	Description
TcpipProcName	z/OS TCP/IP stack name.

CIM_PortImplementsEndpoint

This class associates a network port with its installed network protocols. Currently, only IP protocols defined for Ethernet ports are returned.

The z/OS specific subclass is **IBMzOS_NetworkPortImplementsIPEndpoint**.

CIM_IPProtocolEndpoint

This class represents the installed IP protocols. For z/OS, all IPv4 addresses configured to the TCP/IP stacks on the MVS image are supported.

The z/OS specific subclass is **IBMzOS_IPProtocolEndPoint**.

The following properties are common for @server:

Property	Description
string Caption	Always set to <i>"IBMzOS Protocol Endpoint for IP"</i> .
string Description	Always set to <i>"A communication point to send and receive data. This class is dedicated to relate IP interfaces to Logical Networks"</i> .

Property	Description
string ElementName	Same as Name.
string Name [key]	The unique name of the protocol endpoint, constructed according to the template in NameFormat.
uint16 EnabledState	Returns whether the protocol endpoint is active or not.
string NameFormat	Describes the format of the name property. For z/OS, this is always set to "TCPIPPROCNAME_TYPE_DEVICE_IPADDR(_ETH)".
string IPv4Address	The IPv4 IP address.
string IPv6Address	Not yet supported for z/OS instrumentation.
string SubnetMask	The IPv4 IP subnet mask.
uint16 IPVersionSupport	Always returns 1 (=IPv4 only).

IBMzOS_IPProtocolEndpoint

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

Property	Description
TcpipProcName	z/OS TCP/IP stack name.

CIM_SystemDevice association

This class associates a **ComputerSystem** with the instrumented network ethernet ports.

The z/OS specific subclass is **IBMzOS_CSNetworkPort**.

Chapter 5. Messages and logs

This chapter contains the following sections with information on which logs and messages the z/OS CIM server provides to users:

- “Logging”
- “Checking the CIM server logs”
- “CIM server messages”

Logging

The CIM server writes logging information onto the system console, and also into log files in the `/var/wbem/logs` HFS directory. Those files are written with ASCII encoding.

There are four files:

- `PegasusError.log` (contains CIM server error information)
- `PegasusStandard.log` (contains general logging information)
- `PegasusDebug.log` (contains debugging information)
- `PegasusTrace.log` (contains mostly authentication issues)

For information on how to change the default logging behavior of the CIM server, refer to “CIM server configuration” on page 15.

Checking the CIM server logs

In case of an error, check the various available log files listed in “Logging.” You can set the log level using the `cimconfig` command, for example:

```
cimconfig -s logLevel=TRACE
```

You find a description of the supported log levels in Table 2 on page 15.

CIM server messages

Messages are written into the appropriate logs and also displayed at the z/OS console.

All messages while using the CIM server are part of the underlying OpenPegasus code. This section documents only those messages that are specific while using the CIM server on z/OS, together with explanation, system action, (system) programmer and user response.

All other OpenPegasus messages are wrapped by one of the following generic z/OS messages. For information on these OpenPegasus messages, refer to the *OpenPegasus Administrator's Guide*.

CFZ00001I	for INFORMATION log messages
CFZ00002W	for WARNING log messages
CFZ00004E	for SEVERE and FATAL log messages

z/OS-specific messages

CFZ02001W Request UserID *user_id* does not have READ permission to profile CIMSERV CL(WBEM).

Explanation: The userID requesting a CIM operation using a local connection is not permitted to use the CIM server.

System Action: The CIM request is not executed, but answered with CIM_ERR_ACCESS_DENIED.

System Programmer Response: Permit the user to execute CIM requests by giving the user READ access to profile CIMSERV CL(WBEM).

User Response: Contact your system programmer to permit your userID to execute CIM requests and afterwards repeat your request.

CFZ02002W Request UserID *user_id* doesn't have READ permission to profile CIMSERV CL(WBEM).

Explanation: The userID requesting a CIM operation using a remote connection is not permitted to use the CIM server.

System Action: The CIM request is not executed, but answered with CIM_ERR_ACCESS_DENIED.

System Programmer Response: Permit the user to execute CIM requests by giving the user READ access to profile CIMSERV CL(WBEM).

User Response: Contact your system programmer to permit your userID to execute CIM requests and afterwards repeat your request.

CFZ05203W Authorization failed: UserID *user_id* does not have CONTROL permission to profile CIMSERV CL(WBEM).

Explanation: The userID requesting an administrative task, for example, *cimconfig* or *cimprovider*, does not have the required permission to execute.

System Action: The CIM request is not executed, but answered with CIM_ERR_ACCESS_DENIED.

System Programmer Response: Permit the user to execute administrative CIM tasks by giving him CONTROL permission to profile CIMSERV in class WBEM.

User Response: Contact your system programmer for permission to execute administrative CIM tasks.

CFZ12500E Not loading dynamic load library *library_name* due to missing program control flag.

Explanation: The CIM server runs on a system with

Enhanced Security and thus does not load dynamic libraries which are not audited by a system programmer.

System Action: Does not load the named dynamic library.

System Programmer Response: Set the program control flag on the dynamic library using the USS command `extattr +p <filename>`.

User Response: Contact a system programmer to audit the dynamic library and set the program control flag.

CFZ12501E Security Profile CIMSERV in CLASS WBEM must be defined. Ending CIM server.

Explanation: The CIM server detected an incomplete security setup.

System Action: CIM server startup is stopped.

System Programmer Response: Complete the security setup by defining the profile CIMSERV in class WBEM. Refer to "Setting up security for the CIM server" on page 9 for further details.

User Response: Contact your system programmer.

CFZ12502E CIM server Userid requires either READ access to BPX.SERVER or must be UID 0. Ending CIM server.

Explanation: The userID under which you are running the CIM server must have read access to the security profile BPX.SERVER or, if BPX.SERVER is not defined on your system, must be a privileged user.

System Action: CIM server is stopped.

System Programmer Response: Permit the user to run the CIM server by either giving the user read access to profile BPX.SERVER, or, if not running in an Enhanced Security environment, make the user a privileged user (`uid = 0`).

User Response: Contact your system programmer.

CFZ12503E CIM server address space dirty due to loading from a not program controlled load library. Ending CIM server.

Explanation: The CIM server did load a dynamic library that is not program controlled. Either the security setup is not complete or a dynamic library has been changed without a system programmer's audit.

System Action: CIM server is stopped.

Programmer Response: Check all dynamic libraries for their program control flag and ensure that no

library changed. Make sure the Language Environment libraries SCEERUN and SCEERUN2 are program controlled.

User Response: Contact your system programmer.

CFZ12504E CIM server does not have appropriate privileges to check SAF security environment. Ending CIM server.

Explanation: The userID under which you are running the CIM server must have read access to the security profile BPX.SERVER or, if BPX.SERVER is not defined on your system, must be a privileged user.

System Action: CIM server is stopped.

System Programmer Response: Permit the user to run the CIM server environment by either giving the user read access to profile BPX.SERVER or, if not running in an Enhanced Security environment, make the user a privileged user (uid = 0).

User Response: Contact your system programmer.

CFZ12505E CIM server userID requires either READ access to BPX.SERVER or must be UID 0. Ending CIM server.

Explanation: The userID under which you are running the CIM server must have read access to the security profile BPX.SERVER or, if BPX.SERVER is not defined on your system, must be a privileged user.

System Action: CIM server is stopped.

System Programmer Response: Permit the user to run the CIM server environment by either giving the user read access to profile BPX.SERVER or, if not running in an Enhanced Security environment, make the user a privileged user (uid = 0).

User Response: Contact your system programmer.

CFZ12506E CIM server address space dirty due loading from a not program controlled load library. Ending CIM server.

Explanation: The CIM server did load a dynamic library that is not program controlled. Either the security setup is not complete or a dynamic library has been changed without a system programmer's audit.

System Action: CIM server is stopped.

Programmer Response: Check all dynamic libraries for their program control flag and ensure that no library changed. Make sure the Language Environment libraries SCEERUN and SCEERUN2 are program controlled.

User Response: Contact your system programmer.

CFZ12507W CIM server does not have surrogate for client userid *user_id*

Explanation: A request sent from the userID could not be executed. The CIM server does not have access to act as surrogate for the requesting userID.

System Action: The user request is ignored and an error message is sent to the client.

System Programmer Response: Permit the CIM server userID to act as surrogate for the client user. To achieve this, in RACF read access to profile BPX.SRV.<client userid> is required.

User Response: Contact your system programmer.

CFZ12508W Failure *error_number* deleting thread security

Explanation: The CIM server was not able to delete the thread level security built for a specific request.

System Action: None.

System Programmer Response: None.

User Response: None.

CFZ12509E The CIM server userID requires either READ access to BPX.SERVER or must be UID 0. Stopping CIM server startup.

Explanation: The userID under which you are trying to start the CIM server must have read access to the security profile BPX.SERVER or, if BPX.SERVER is not defined on your system, must be a privileged user.

System Action: CIM server startup is stopped.

System Programmer Response: Permit the user to run the CIM server by either giving the user read access to profile BPX.SERVER or, if not running in an Enhanced Security environment, make the user a privileged user (uid = 0).

User Response: Contact your system programmer.

CFZ12510E CIM server address space dirty due to loading from a not program controlled load library. Stopping CIM server startup.

Explanation: The CIM server did load a dynamic library that is not program controlled during startup. Probably the security setup is not complete or a dynamic library has been changed without a system programmer's audit.

System Action: CIM server startup is stopped.

Programmer Response: Check all dynamic libraries for their program control flag and ensure that no library changed. Make sure the Language Environment libraries SCEERUN and SCEERUN2 are program controlled.

User Response: Contact your system programmer.

CFZ12511E CIM server does not have appropriate privileges to check SAF security environment. Stopping CIM server startup.

Explanation: The userID under which you are trying to start the CIM server must have read access to the security profile BPX.SERVER or, if BPX.SERVER is not defined on your system, must be a privileged user.

System Action: The CIM server does not start.

System Programmer Response: Permit the user to run the CIM server environment by either giving the user read access to profile BPX.SERVER or, if not running in an Enhanced Security environment, make the user a privileged user (uid = 0).

User Response: Contact your system programmer.

CFZ12512E Security Profile CIMSERV in CLASS WBEM must be defined. Stopping CIM server startup.

Explanation: The CIM server detected an incomplete security setup on startup.

System Action: CIM server startup is stopped.

System Programmer Response: Complete the security setup by defining the profile CIMSERV in class WBEM. Refer to "Setting up security for the CIM server" on page 9 for further details.

User Response: Contact your system programmer.

CFZ12513E The CIM server userID requires CONTROL access to security profile CIMSERV in CLASS WBEM. Stopping CIM server startup.

Explanation: The CIM server userID requires CONTROL access to security profile CIMSERV in CLASS WBEM.

System Action: CIM server startup is stopped.

System Programmer Response: Permit the CIM server userID to execute administrative CIM tasks by giving it CONTROL permission to profile CIMSERV in class WBEM. Refer to "Setting up security for the CIM server" on page 9 for further details.

User Response: Contact your system programmer.

CFZ14208W Received error: *error_number* while binding the internal socket.

Explanation: The CIM server failed to bind a socket (either HTTP or HTTPS). This can have several causes. In most cases, another service is already running and using the port, or the ports might be reserved.

System Action: CIM server does not start.

System Programmer Response: Either free the ports for HTTP(5988) or HTTPS(5989) connection by shutting down the service using them or reconfigure the CIM server port settings. It is recommended to use the default ports (5988 and 5989) for the CIM server.

User Response: Contact your system programmer.

Appendix A. Sample CIM request

The provided sample shows an XML request for an EnumerateInstances request for class IBMzOS_ComputerSystem and the related XML response. See the *CIM Operations over HTTP* and *Representation of CIM in XML* specifications available from the DMTF Published Documents website.

Request:

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
    <SIMPLEREQ>
      <IMETHODCALL NAME="EnumerateInstances">
        <LOCALNAMESPACEPATH>
          <NAMESPACE NAME="root"></NAMESPACE>
          <NAMESPACE NAME="cimv2"></NAMESPACE>
        </LOCALNAMESPACEPATH>
        <IPARAMVALUE NAME="ClassName">
          <CLASSNAME NAME="IBMzOS_ComputerSystem"/>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="DeepInheritance">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="LocalOnly">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="IncludeQualifiers">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="IncludeClassOrigin">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
      </IMETHODCALL>
    </SIMPLEREQ>
  </MESSAGE>
</CIM>
```

Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
    <SIMPLERSP>
      <IMETHODRESPONSE NAME="EnumerateInstances">
        <IRETURNVALUE>
          <VALUE.NAMEDINSTANCE>
            <INSTANCENAME CLASSNAME="IBMzOS_ComputerSystem">
              <KEYBINDING NAME="CreationClassName">
                <KEYVALUE VALUETYPE="string">IBMzOS_ComputerSystem</KEYVALUE>
              </KEYBINDING>
              <KEYBINDING NAME="Name">
                <KEYVALUE VALUETYPE="string">BOEMT11.boeb.de.ibm.com</KEYVALUE>
              </KEYBINDING>
            </INSTANCENAME>
            <INSTANCE CLASSNAME="IBMzOS_ComputerSystem" >
              <PROPERTY NAME="Caption" TYPE="string">
                <VALUE>IBM z/OS Computer System</VALUE>
              </PROPERTY>
              <PROPERTY NAME="Description" TYPE="string">
                <VALUE>This is an IBMzOS_ComputerSystem</VALUE>
              </PROPERTY>
              <PROPERTY NAME="ElementName" TYPE="string">
```

```

        <VALUE>BOEMT11.boeb.de.ibm.com</VALUE>
    </PROPERTY>
    <PROPERTY NAME="InstallDate" TYPE="datetime">
    </PROPERTY>
    <PROPERTY.ARRAY NAME="OperationalStatus" TYPE="uint16">
        <VALUE.ARRAY>
            <VALUE>0</VALUE>
        </VALUE.ARRAY>
    </PROPERTY.ARRAY>
    <PROPERTY.ARRAY NAME="StatusDescriptions" TYPE="string">
    </PROPERTY.ARRAY>
    <PROPERTY NAME="Status" TYPE="string">
    </PROPERTY>
    <PROPERTY NAME="EnabledState" TYPE="uint16">
        <VALUE>2</VALUE>
    </PROPERTY>
    <PROPERTY NAME="OtherEnabledState" TYPE="string">
        <VALUE>NULL</VALUE>
    </PROPERTY>
    <PROPERTY NAME="RequestedState" TYPE="uint16">
        <VALUE>2</VALUE>
    </PROPERTY>
    <PROPERTY NAME="EnabledDefault" TYPE="uint16">
        <VALUE>2</VALUE>
    </PROPERTY>
    <PROPERTY NAME="TimeOfLastStateChange" TYPE="datetime">
    </PROPERTY>
    <PROPERTY NAME="CreationClassName" TYPE="string">
        <VALUE>IBMzOS_ComputerSystem</VALUE>
    </PROPERTY>
    <PROPERTY NAME="Name" TYPE="string">
        <VALUE>BOEMT11.boeb.de.ibm.com</VALUE>
    </PROPERTY>
    <PROPERTY NAME="PrimaryOwnerName" TYPE="string">
    </PROPERTY>
    <PROPERTY NAME="PrimaryOwnerContact" TYPE="string">
    </PROPERTY>
    <PROPERTY.ARRAY NAME="Roles" TYPE="string">
        <VALUE.ARRAY>
            <VALUE>Unknown</VALUE>
        </VALUE.ARRAY>
    </PROPERTY.ARRAY>
    <PROPERTY NAME="NameFormat" TYPE="string">
        <VALUE>IP</VALUE>
    </PROPERTY>
    <PROPERTY.ARRAY NAME="OtherIdentifyingInfo" TYPE="string">
    </PROPERTY.ARRAY>
    <PROPERTY.ARRAY NAME="IdentifyingDescriptions" TYPE="string">
    </PROPERTY.ARRAY>
    <PROPERTY.ARRAY NAME="Dedicated" TYPE="uint16">
        <VALUE.ARRAY>
            <VALUE>0</VALUE>
        </VALUE.ARRAY>
    </PROPERTY.ARRAY>
    <PROPERTY.ARRAY NAME="OtherDedicatedDescriptions" TYPE="string">
    </PROPERTY.ARRAY>
    <PROPERTY NAME="ResetCapability" TYPE="uint16">
        <VALUE>5</VALUE>
    </PROPERTY>
    <PROPERTY.ARRAY NAME="PowerManagementCapabilities" TYPE="uint16">
    </PROPERTY.ARRAY>
    <PROPERTY NAME="MachineType" TYPE="string">
        <VALUE>2084</VALUE>
    </PROPERTY>
    <PROPERTY NAME="Model" TYPE="string">
        <VALUE>314</VALUE>
    </PROPERTY>

```

```

    <PROPERTY NAME="SerialNumber" TYPE="string">
      <VALUE>00000000000016F7A</VALUE>
    </PROPERTY>
    <PROPERTY NAME="LPARName" TYPE="string">
      <VALUE></VALUE>
    </PROPERTY>
    <PROPERTY NAME="VMGuestID" TYPE="string">
      <VALUE>MT11</VALUE>
    </PROPERTY>
    <PROPERTY NAME="CPUID" TYPE="string">
      <VALUE>026F7A2084</VALUE>
    </PROPERTY>
  </INSTANCE>
</VALUE.NAMEDINSTANCE>
</IRETURNVALUE>
</IMETHODRESPONSE>
</SIMPLERSP>
</MESSAGE>
</CIM>

```

Appendix B. Classes and their providers

OS management classes

The following **OS management** classes are implemented by providers of **z/OS CIM**:

Base classes:

- IBMzOS_ComputerSystem
- IBMzOS_OperatingSystem
- IBMzOS_OSProcess
- IBMzOS_RunningOS
- IBMzOS_UnixProcess
- IBMzOS_Process

Processor classes:

- IBMzOS_CSProcessor
- IBMzOS_Processor

File System classes:

- IBMzOS_HostedFileSystem
- IBMzOS_NFS
- IBMzOS_UnixLocalFileSystem

The following **OS management** classes are implemented by providers of **z/OS Communications Server**:

Network classes:

- IBMzOS_CSNetworkPort
- IBMzOS_EthernetPort
- IBMzOS_IPProtocolEndpoint
- IBMzOS_NetworkPortImplementsIPEndpoint

OS monitoring classes

The following **OS monitoring** classes are implemented by **z/OS RMF** monitoring providers:

- IBMzOS_BaseMetricValue
- IBMzOS_BaseMetricDefinition
- IBMzOS_MetricForME
- IBMzOS_MetricDefForME
- IBMzOS_MetricInstance
- IBMzOS_Channel

Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

Notices

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

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

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

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

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
Mail Station P300
2455 South Road
Poughkeepsie New York 12601-5400
U.S.A.

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

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

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

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

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.

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

Programming Interface Information

This book is intended to help the customer to use the Common Information Model to write system management applications for z/OS systems.

The book also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of CIM.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries:

IBM ???

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

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 AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Index

A

access
 UPDATE 12
accessibility 55
authentication 10, 13
 basic 13
authorization 10, 12

B

Base classes 29
 OS management 30
basic authentication 13
BPX.SERVER 12
BPX.SRV 12

C

CDT 10
CDTINFO 11
CFZCIM 7
 start 14
 started task 14
 stop 15
CIM v, 3
 introduction 3
 user access 13
CIM client application 4
CIM operations 23
CIM server 4
 configuration 15
 customizing 7
 how to work with 7
 setting up 7
 setting up security 9
 starting and stopping 14
CIM server runtime environment 12
CIM server runtime environment
 security 9
CIM_ComputerSystem 31
CIM_EthernetPort 41
CIM_HostedFileSystem 38
CIM_IPProtocolEndpoint 42
CIM_LocalFileSystem 38
CIM_OperatingSystem 32
CIM_OSProcess 35
CIM_PortImplementsEndpoint 42
CIM_Process 33
CIM_Processor 36
CIM_RemoteFileSystem 39
CIM_RunningOS 33
CIM_SystemDevice association 36, 43
cimconfig 20
 startup options 15
cimmof 19
cimmofl 19
cimprovider 22
CIMSERV 13
CIMSERV profile 11
cimserver command 14

cimserver.env 7, 8
class descriptor table 10
class SURROGAT 12
client application 4
client monitoring application 25
command
 cimconfig
 startup options 15
 cimserver 14
command-line utilities 18
commands 18
Common Information Model v, 3
 introduction 3
Communications Server 29
configuration properties
 changing current 15
 changing planned 15
customizing
 CFZCIM 7
 CIM server 7
 profile.add 8
 started task 7
 USS shell 8

D

daemon 15
disability 55
Distributed Management Task Force 3
DMTF 3
dynamic CDT 10

E

enableAssociationTraversal 16
enableHttpConnection 16
enableHttpsConnection 16
enableProviderProcesses 16
enableRemotePrivilegedUserAccess 16
Enhanced Security model 10
EnumerateInstanceNames 23
EnumerateInstances 23
environment variables 8
error messages 25
eServer CIM 5
extattr 12

F

file system classes
 OS management 38
File System classes 29

G

GetInstance 23

H

HTTP 13
httpAuthType 16
httpPort 16
HTTPS 13
httpsPort 16

I

IBMzOS_ComputerSystem 31
IBMzOS_EthernetPort 42
IBMzOS_IPProtocolEndpoint 43
IBMzOS_OperatingSystem 32
IBMzOS_OSProcess 35
IBMzOS_Process 33
IBMzOS_RunningOS 33
IBMzOS_UnixLocalFileSystem 39
IBMzOS_UnixProcess 34

K

keyboard 55

L

LIBPATH 8
logdir 16
logging 45
logLevel 16

M

managed object format 18
messageDir 16
MOF 18
monitoring provider 9

N

network classes
 OS management 40
Network classes 29

O

Open Group 4
OpenPegasus 4
OS management Base classes 29
OS management classes 30
OS management file system classes 38
OS management File System classes 29
OS management network classes 40
OS management Network classes 29
OS management processor classes 36
OS management Processor classes 29
OSBASE_TRACE 8
OSBASE_TRACE_FILE 8

P

- PATH 9
- PEGASUS_HOME 8
- privileged user 12
- problem determination 25
- processor classes
 - OS management 36
- Processor classes 29
- profile
 - CIMSERV 11, 13
- profile.add 7, 8
- program control flag
 - extattr 12
- provider 4
- providerDir 17
- pszos
 - sample client application 24

R

- repositoryDir 17
- repositoryIsDefaultInstanceProvider 17
- RMF monitoring provider 9
- RMF_CIM_BENCH 9
- RMF_CIM_HOST 9
- RMF_CIM_PORT 9
- RMF_CIM_TRACE 9
- RMF_CIM_TRACE_FILE 9
- runtime environment
 - CIM server 12
- runtime environment security
 - CIM server 9

S

- sample client application 24
- SCEERUN 12
- SCEERUN2 12
- setting up
 - CIM server 7
- setting up security
 - CIM server 9
- setup
 - z/OS based authorization 12
- shortcut keys 55
- shutdownTimeout 17
- slp 17
- started task
 - CFZCIM 14
- starting the CIM server 14
- STDERR 7
- STDOUT 7
- stopping the CIM server 14
- SURROGAT class 12

T

- tempLocalAuthDir 17
- traceComponents 17
- traceFilePath 18
- traceLevel 18
- troubleshooting 25
- TTLS 13

U

- UPDATE access 12
- user access to CIM 13

W

- WBEM v, 3
- Web Base Enterprise Management v
- Web Based Enterprise Management 3

Z

- z/OS based authorization
 - setup 12

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

z/OS

Common Information Model User's Guide

Publication No. SC33-7998-00

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

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

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

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



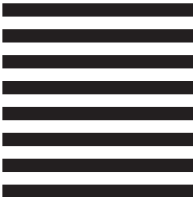
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5694-A01, 5655-G52

Printed in USA

SC33-7998-00



Spine information:



z/OS

z/OS V1R7.0 Common Information Model User's
Guide

SC33-7998-00