

IBM TotalStorage



Common Information Model Agent Installation and Configuration Guide

for the IBM Enterprise Storage Server

IBM TotalStorage



Common Information Model Agent Installation and Configuration Guide

for the IBM Enterprise Storage Server

Note:

Before using this information and the product it supports, read the information in "Notices" on page 143.

First Edition (February 2003)

© Copyright International Business Machines Corporation 2002. All rights reserved.

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

Contents

About this guide	v
Who should use this guide	v
Conventions used in this guide	v
How to send your comments	vi
 Chapter 1. Introduction	 1
CIM Agent overview	1
CIM Agent components	2
CIM overview	3
CIM Agent installation requirements	4
Hardware	4
Workstation space	4
Software	4
CIM Agent installation methods	5
CIM Agent security	5
 Chapter 2. CIM Agent for AIX	 7
Installation overview for AIX	7
Installing the ESS CLI for AIX	7
Mounting the CD	9
Installing the CIM Agent on AIX in graphical mode	10
Installing the CIM Agent on AIX in unattended (silent) mode	16
Verifying the CIM Agent installation for AIX	19
Configuring the CIM Agent for AIX	20
Configuring the CIM Agent to run in unsecure mode	21
Running the CIM Agent on AIX	22
Removing the CIM Agent for AIX	23
Removing in graphical mode	24
Removing in unattended (silent) mode	27
 Chapter 3. CIM Agent for Linux	 29
Installation overview for Linux	29
Installing the ESS CLI for Linux	29
Installing the CIM Agent on Linux in graphical mode	31
Installing the CIM Agent on Linux in unattended (silent) mode	38
Verifying the CIM Agent installation for Linux	42
Configuring the CIM Agent for Linux	42
Configuring the CIM Agent to run in unsecure mode on Linux	44
Running the CIM Agent on Linux	44
Removing the CIM Agent for Linux	46
 Chapter 4. CIM Agent for Windows	 51
Installation overview for Windows	51
Installing the ESS CLI for Windows	51
Installing the CIM Agent on Windows in graphical mode	53
Installing the CIM Agent on Windows in unattended (silent) mode	60
Verifying the CIM Agent Windows installation	63
Configuring the CIM Agent for Windows	63
Configuring the CIM Agent to run in unsecure mode	64
Verifying connection to the ESS	65
Removing the CIM Agent for Windows	66
 Chapter 5. CIM Agent commands	 73

Conventions used in this chapter	73
Syntax diagrams	73
Special characters	75
Emphasis	75
Configuration commands	76
setuser	76
setdevice	79
Operational commands	84
startcimom	84
stopcimom	85
Utility commands	85
mkcertificate	85
slpd	86
verifyconfig	86
Appendix A. CIM model component definitions	89
Appendix B. CIM Agent communication information	91
CIM Agent communication concepts	91
CIM Agent Communication Methods	92
Error codes returned by the CIMOM	102
Appendix C. ESS CIM Agent Information	105
ESS CIM Agent class definitions quick reference	105
ESS CIM Agent Class Definitions.	108
ESS class definition schemas	137
Notices	143
Trademarks.	144
Java Compatibility Logo	145
Glossary	147
Index	157

About this guide

This publication introduces the IBM® TotalStorage™ Common Interface Model (CIM) concepts and components and provides instructions for installing and configuring the CIM Agent on the following operating systems:

- IBM Advanced Interactive Executive (AIX®)
- Linux
- Microsoft® Windows® 2000

You can install the CIM Agent on a host server or on a workstation within a network.

This publication also provides descriptions of the commands that you use during the installation and configuration tasks.

Reference material in the appendixes includes the following information that might assist you in writing your CIM-based applications:

- CIM model component definitions
This appendix describes the elements, the namespace, and the object name for the CIM model.
- CIM Agent communication information
This appendix describes the concepts and methods for communication between the ESS CIM Agent and the client application and lists error codes returned by the CIMOM
- ESS CIM Agent Information
The ESS CIM Agent uses the classes described in this section to manage its model of the ESS. Also included is a visual representation of the CIM agent architecture for the ESS.

Who should use this guide

This publication is for system administrators and system programmers, or whoever is responsible for installing and configuring the CIM Agent. This publication assumes that you understand the general concepts of the operating system and Internet capabilities for your enterprise.

Reference material for application programmers who write CIM-based applications is in the appendixes.

Conventions used in this guide

This guide uses bold, italic, and other typographic styles to highlight various types of information in the text. See Table 1 for a list of conventions.

Table 1. Documentation conventions

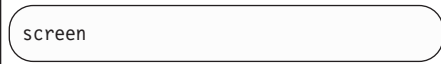
bold	Denotes a command, user ID, password, or labels on buttons.
<i>italics</i>	Denotes titles of other manuals or books.
monospace	Denotes a Web address that needs to be typed in. Also denotes an example.
	Used to show the screen output you should see.

Table 1. Documentation conventions (continued)

Note	
	Introduces special considerations about the current topic.
(>) greater than and (<) less than	Shows the optional and substitution parameters in the description of commands. The explanation of the optional or substitution parameters are included following the word "Where."

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book, use one of the following methods to provide feedback.

- Send your comments by electronic mail (e-mail) to:
 - IBMLink™ from US: starpubs@us.ibm.com
 - IBMLink from Canada: STARPUBS at TORIBM
 - IBM Mail Exchange: USIB3VVD at IBMMAIL
 - Internet: starpubs@us.ibm.com

Be sure to include the name of the book, the part number of the book, version, and product name. Also, if applicable, include the specific location of the text about which you are commenting (for example, a page number or a table number).

- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to:

IBM Corporation,
RCF Processing Department
M86/050-3
5600 Cottle Road
San Jose, CA, 95193-0001

Chapter 1. Introduction

This chapter provides the following information about the Common Information Model (CIM) Agent, CIM standards, and CIM Agent installation:

- CIM Agent overview
- CIM Agent components
- CIM concepts
- CIM Agent installation requirements
- CIM Agent installation methods
- CIM Agent security

CIM Agent overview

The Common Information Model (CIM) Agent provides a means by which a device can be managed by common building blocks rather than proprietary software. If a device is CIM-compliant, software that is also CIM-compliant can manage the device. Vendor applications can benefit from adopting the common information model because they can manage CIM-compliant devices in a common way, rather than using device-specific programming interfaces. Using CIM, you can perform tasks in a consistent manner across devices and vendors.

The CIM agent consists of the components shown in Figure 1 on page 2. The main components are the CIM object manager (CIMOM), the service location protocol (SLP) and the device provider. A device can be a storage server such as your IBM Enterprise Storage Server™ (ESS). The SLP is a directory service that a client application calls to locate the CIM Object Manager. The client application and the CIMOM communicate through CIM Messages. The CIMOM and device provider communicate through method calls made from the CIMOM to the provider. The device provider communicates with the device through proprietary calls.

The CIMOM supports the following specifications and standards:

- The Distributed management Task Force (DMTF) Specification for CIM Operations over HTTP
- The Storage Networking Industry Association (SNIA) Storage Management Initiative (SMI) Specification
- CIM Schema 2.7 for Managing a Storage Array

Conformance to these specifications allows the CIM Agent to act as an open-system standards interpreter, allowing other CIM-compliant storage resource management applications (IBM and non-IBM) to interoperate with each other.

When you have installed, configured, and enabled the CIM Agent on a host server or an administrator's workstation within your network, that host server or workstation can communicate with your ESS through the CIM Agent. However, installing the CIM Agent is only the first of several tasks that you need to perform before data on your ESS can be managed by CIM-compliant software applications. See the appendixes to this guide for information about these additional tasks and about the ESS CIM API.

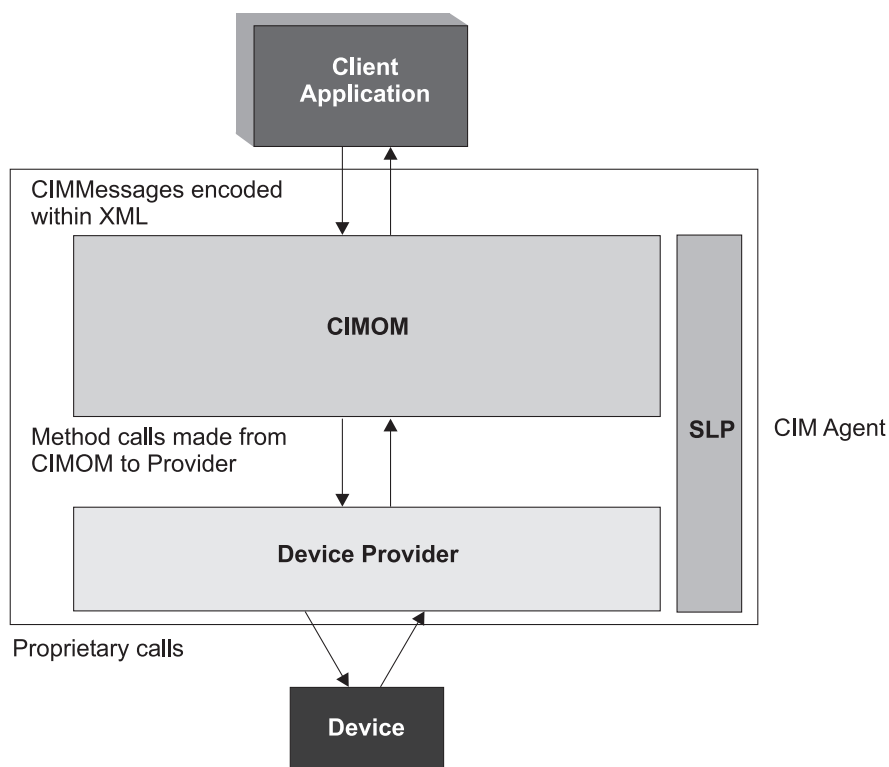


Figure 1. How the CIM Agent works

CIM Agent components

The following list describes the CIM Agent components:

client application	A storage management API that initiates a request to a device or a data storage server such as an ESS
CIM Agent	An agent that interprets open-system data as it is transferred between the API and a device or a storage server.
service location protocol (SLP)	A directory service that a client application calls to locate the CIM Object Manager.
CIM object manager (CIMOM)	A common conceptual framework for data management. Receives, validates, and authenticates client application requests, then directs requests to the appropriate functional component or to a device provider.
Provider	A device-specific handler that receives client application requests that are destined for its device or storage server.
Device (also known as a storage server or an ESS)	The final destination of a client application request, and the processor of the request.

CIM overview

The common information model (CIM) is an approach to the management of systems and networks. The CIM provides a common conceptual framework applicable to all areas of management including systems, applications, databases, networks, and devices. The CIM specification provides the language and the methodology used to describe management data.

The CIM defines a set of classes with properties and associations that provide a well-understood conceptual framework within which it is possible to organize the available information about the managed environment. CIM Schema 2.7 for Managing a Storage Array, provides information about enabling management applications to manage data in a common way.

The CIM standard and the DMTF Specification provide information about Web-based enterprise management (WBEM) operations over HTTP. For additional information about CIM Operations over HTTP, visit the following Web site:

http://www.dmtf.org/download/spec/xmls/CIM_HTTP_Mapping10.htm

A client application finds the location of the CIMOM by calling an SLP directory service. When the CIMOM first starts, it registers itself to the SLP and provides information about its location (IP address and port) and the type of service it provides. After obtaining this information, the client application opens direct communication with the CIMOM.

A client sends requests to a CIMOM in the context of a CIM model. The model is defined by the CIM schema and loaded into the repository of the CIMOM. Figure 2 shows how the schema is loaded into the data store of the CIMOM. The MOF compilation and creation of the data store is handled automatically during installation.

As requests arrive, the CIMOM validates and authenticates each request. Requests are either directed to the appropriate functional component of the CIMOM or directed to a device-specific handler called a provider.

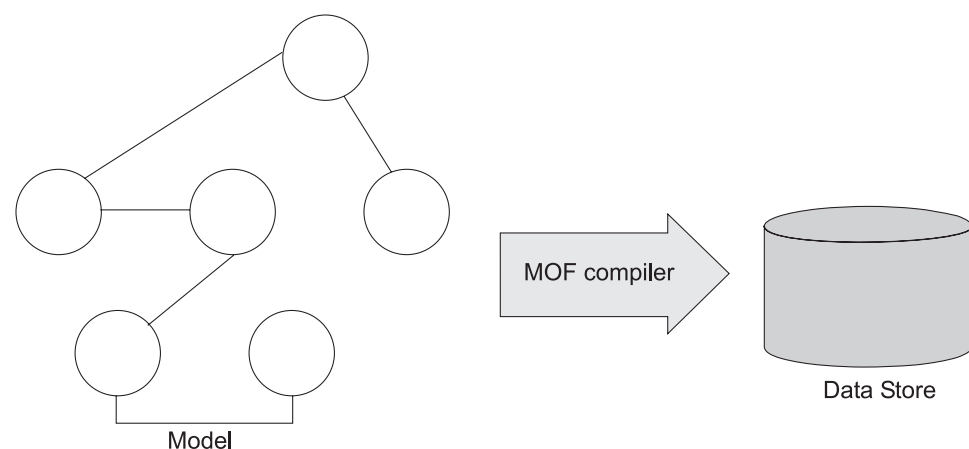


Figure 2. The MOF compiler stores the model in the CIMOM data store.

A provider makes device-unique programming interface calls on behalf of the CIMOM to satisfy a client application request. Such requests generally map a CIM request to the propriety programming interface for a device. A request to get an instance of a class or a property of an instance, for example, might be directed to a

Provider and a Provider might make one or many requests of a device using the unique API for the device. Figure 1 on page 2 shows the communication protocol between the device and the client application.

For additional information about CIM operations over HTTP, visit the following Web site:

http://www.dmtf.org/download/spec/xmls/CIM_HTTP_Mapping10.htm

CIM Agent installation requirements

Ensure that your system satisfies the following prerequisites for installing the CIM Agent on a Windows 2000, AIX, or Linux operating system before starting the installation.

- You must install the ESS command-line interface (CLI) on your operating system *before* you install the CIM Agent.

ESS CLI installation is unique for each operating system. See the appropriate chapter in this publication for the instructions for installing the ESS CLI on your operating system:

- Chapter 4, “CIM Agent for Windows”, on page 51
- Chapter 2, “CIM Agent for AIX”, on page 7
- Chapter 3, “CIM Agent for Linux”, on page 29

Note: All prerequisite software must be installed before you start the installation of the CIM Agent. The CIM Agent installation program will check for the existence of the ESS CLI. If the program does not detect the ESS CLI, the installation of the CIM Agent will not complete successfully.

Hardware

The following hardware is required:

- Personal computer, workstation, or server with Intel® Pentium® II or higher processor
- CD-ROM drive
- Video graphics adapter display or better

Workstation space

The following space on your workstation is required:

- 256 megabytes (MB) of random-access memory (RAM) minimum depending on your system configuration
- 100 MB of disk space

Note: You might need to increase the total available disk space on your hard drives if the CIM Agent and other associated products are split between more than one logical drive.

- Up to 50 MB of temporary disk space for installation purposes

Software

The following software is required:

- Windows 2000, AIX, or Linux operating system
- ESS CLI level 2.1.0.0 or later. This software is on the *ESS CLI* CD.
- Common Information Model (CIM) Agent. This software is on the *Common Information Model Agent* CD.

- Transmission Control Protocol/Internet Protocol (TCP/IP)
 - Adobe Acrobat Reader version 4.0 (optional)
- You can download the Adobe Acrobat Reader from the following Web site:
- <http://www.adobe.com/support/downloads/main.html>

CIM Agent installation methods

You can choose to install the CIM Agent in graphical mode with the help of the installation Wizard or in unattended mode (also known as silent mode), which involves customizing a response file and issuing a command. Follow the instructions in the chapter appropriate for your operating system:

- Chapter 4, “CIM Agent for Windows”, on page 51
- Chapter 2, “CIM Agent for AIX”, on page 7
- Chapter 3, “CIM Agent for Linux”, on page 29

CIM Agent security

The CIM Agent can operate in both a secure and unsecure modes.

Secure mode

All requests between the client application and the CIMOM are XML encoded requests sent over Hypertext Transfer Protocol (HTTP) or HTTP over Secure Sockets Layer (SSL). The CIMOM, upon receiving a request, parses the request and processes it. Responses, when they are returned to the client application, are transformed into XML-encoded CIM status and returned in HTTP responses to the client. The default of the CIM Agent is to run in secure mode using SSL.

Unsecure mode

Some vendor software might not be capable of communicating with the CIM Agent in a secure mode. You can still use this vendor software by configuring the CIM Agent to run with only basic user name and password security. See the configuration instructions for your operating system for the instructions for configuring the CIM Agent for this less secure mode.

Chapter 2. CIM Agent for AIX

This chapter includes an overview of the installation process and instructions for installing and configuring the CIM Agent on an IBM AIX® operating system.

Installation overview for AIX

This section provides an overview and instructions for installing and configuring the CIM Agent on the AIX operating system. You should have some knowledge of how to administer AIX operating system before you begin to install the CIM Agent. You should also become familiar with the command explanations that you use to install and configure the CIM Agent. See Chapter 5, “CIM Agent commands”, on page 73 for information about the commands.

The following list of installation and configuration tasks are in the order in which they should be performed:

1. Before you install the CIM Agent on an AIX operating system, check the hardware and software requirements listed in “CIM Agent installation requirements” on page 4.
2. Install the prerequisite ESS CLI 2105 for AIX software as instructed in “Installing the ESS CLI for AIX”.
3. You can choose to install the CIM Agent either in graphical mode with the help of a wizard or in unattended mode (also known as silent mode), which involves customizing a response file and issuing a command.
 - a. If you want to install the CIM Agent in graphical mode, perform the instructions in “Installing the CIM Agent on AIX in graphical mode” on page 10.
 - b. If you want to install the CIM Agent in unattended mode, perform the instructions in “Installing the CIM Agent on AIX in unattended (silent) mode” on page 16.
4. Configure the CIM Agent by performing the instructions in “Configuring the CIM Agent for AIX” on page 20. You might want to revisit this section in the future as you add, change, or delete CIMOM authentication and ESS information.
5. Enable the CIM Agent by performing the instructions in “Running the CIM Agent on AIX” on page 22.
6. For instructions about removing the CIM Agent, see “Removing the CIM Agent for AIX” on page 23. You only need to perform this optional task if you get errors during installation verification or if the CIM Agent did not set the environment variables.

Installing the ESS CLI for AIX

Before you install the CIM Agent, you must install the ESS CLI, which is sometimes referred to as the *CLI for ESS 2105* or the *Storage Management CLI*.

Steps:

Perform the following steps to install the ESS CLI for AIX:

1. Install the ESS CLI before you install CIM Agent.

The ESS CLI must be installed first because the CIM Agent sets the path information in batch commands for you based on the ESS CLI location on the

disk. The CIM Agent installation wizard checks your system for the existence of the ESS CLI and the wizard stops if the ESS CLI is not installed.

2. Select the *ESS CLI* CD to install the ESS CLI.
3. Select the correct software from the *ESS CLI* CD.

The *ESS CLI* CD contains two products with similar names. Insert the *ESS CLI* CD, but *do not mount* the CD device. Instead, type the **smitty install_update** command and choose `ibm2105esscli.rte` from the available software on the *ESS CLI* CD.

Attention: Do *not* install both the `ibm2105esscli.rte` and `ibm2105cli.rte` at the same time.

4. Verify that the ESS CLI is installed.

Type the following command to review the installed programs on your AIX operating system. The level of the software should be 2.1.0.8 or higher.

```
# lsllpp -l "ibm2105*"
```

If the ESS CLI is installed, a message similar to the following is displayed.

```
ibm2105esscli.rte 2.1.0.9 COMMITTED IBM 2105 ESS Command Line
```

5. Verify that the ESS CLI is operational and that all the environment variables have been set.

The ESS CLI does not set the system path. You need to find where the `esscli` is installed. If you do not know where the `esscli` is installed, type the following command to find out:

```
# lsllpp -f ibm2105esscli.rte | grep esscli
```

The following example output indicates that the `esscli` is installed at level 2.1.0.14 in `/usr/opt/ibm2105cli`.

```
ibm2105esscli.rte 2.1.0.14 /usr/opt/ibm2105cli/esscli
```

Type the following command from a Command Prompt window to see a listing of the `esscli` commands:

```
# /usr/opt/ibm2105cli/esscli
```

The following output is an example of a listing of the `esscli` commands.


```

# /usr/opt/ibm2105cli/esscli

Thu Oct 17 10:15:50 PDT 2002 IBM ESSCLI 2.1.0.8

esscli -a AccessFile | -u Username -p Password
-s ServerIpAddr [-b BackupServer]
[-v] [-fmt "Format"] [-nohdr] [-noss1]
list Log
list Task
list Server
list DiskGroup -d "ess=EssId"
list PAV -d "ess=EssId volume=VolId | lss=LssId"
list Port -d "ess=EssId"
list Volume -d "ess=EssId [lss=LssId] [voltype=VolType]"
list VolumeSpace -d "ess=EssId"
list HostConnection -d "ess=EssId"
list PPRCPaths -d "ess=EssId [srcLss=LssId]"
list VolumeAccess [-d "init=WwName | host=HostName |
    ess=EssId volume=ALL | volume=UNASSIGNED | volume=VolId1,VolId2,..."]
create HostConnection -d "init=WwName ess=EssId host=HostName
    profile=ProfileName [ports=PortId1,PortId2,...]"
create Volume -d "ess=EssId [voltype=VolType] [qty=Quantity] cap=Gigabytes"
    [cyls=Cylinders lss=LssId1,LssId2,... | vs=VSId1,VSId2,..."
    [redundancy=RAID5|RAID10] [placement=SEQ|SPREAD]"
create VolumeAccess -d "init=WwName ess=EssId volume=VolId1,VolId2,..."
delete HostConnection -d "init=WwName ess=EssId"
delete VolumeAccess -d "init=WwName ess=EssId
    volume=ALL | volume=VolId1,VolId2,..."
set Volume -d "ess=EssId volume=VolId label=VolLabel"
set Port -d "ess=EssId port=PortId topology=FCAL|P2P|UNDEFINED"
set HostConnection -d "init=WwName ess=EssId newinit=WwName"
set HostConnection -d "init=WwName ess=EssId host=NickName"
set HostConnection -d "init=WwName ess=EssId ports=All | ports=PortId1,PortId2,
    ."
show Task -d "name=TaskName"
#

```

Mounting the CD

This section provides instructions about how to mount a CD.

Steps:

Perform the following steps to mount the *Common Information Model Agent* CD.

Note: You must mount the *Common Information Model Agent* CD for both a graphical and unattended installation.

1. Log on as a user with root authority.
2. Create a mount point or choose an existing mount point.

To create a mount point called /cdrom, type the following command:

```
# mkdir /cdrom
```

3. Type the following command to mount the CD file system at the desired mount point:

```
# mount -o ro -v cdrfs /dev/cd0 /cdrom
```

4. Change the current directory to the mount point for the CD drive in the AIX subdirectory. For example, if the CD was mounted at the /cdrom mount point, type the following command:

```
# cd /cdrom/AIX
```

Installing the CIM Agent on AIX in graphical mode

This section includes the steps to install the CIM Agent in your AIX environment using the graphical mode. You must comply with all prerequisites listed in “CIM Agent installation requirements” on page 4 before you start the installation.

You can choose to install the CIM Agent in graphical mode with the help of an installation wizard or in unattended (silent) mode, which involves customizing a response file and issuing a command. If you want to install the CIM Agent in unattended mode, see “Installing the CIM Agent on AIX in unattended (silent) mode” on page 16, otherwise continue with this section. After the completion of either kind of installation, you will then verify the installation of the CIM Agent in section, “Verifying the CIM Agent installation for AIX” on page 19. Before you install the CIM Agent on AIX, you should check the hardware and software requirements listed in “CIM Agent installation requirements” on page 4.

Steps:

This task will assist you with the installation of the CIM Agent in your AIX environment using the graphical mode.

Perform the following steps to install the CIM Agent on your AIX operating system system:

1. Log on as a user with root authority.
2. You can run the wizard from either the main console or from a remote X server. If you run it from a remote X server, perform the following steps prior to running the wizard:
 - a. Set the DISPLAY variable to *hostname:displaynumber.screennumber* where:

hostname

The host name of the platform on which the X server runs and from which the wizard starts.

displaynumber

Use the number 0 if X server controls more than one keyboard and monitor unit, for instance, a network of X terminals.

screennumber

This specifies which monitor to use in a multiple monitor setup.

```
<hostname>:<displaynumber.screennumber>
```

Note: Because logging on as root user from the AIX main console, automatically specifies the correct settings, you do not need to perform the following steps. However, if you did *not* log on as a root user, you will need to manually specify these settings under the following circumstances:

- 1) If you log on as a nonroot user, switch to the root user (depending on the profile of the root user).
- 2) If you log on using another computer (another UNIX machine or a PC running an X terminal emulator), referred to as an X server. An X server is a graphical machine, acting as a graphical terminal for a UNIX (in this case AIX) computer, through a special protocol. In order to work together, the program running on the AIX operating system must know the host name (or IP

address) and screen number (normally 0) of the machine acting as the X server. This is accomplished by properly setting the environment variable to DISPLAY on the AIX operating system. On its turn, the X server (if it is a UNIX machine), must be informed to accept the applications running on the AIX operating system as X clients. For example, as applications are allowed to use it as an X terminal . This is specified by the command **xhost** (the form **xhost +** allows any machine to use this X server, or a more restrictive one **xhost aix_name_or_ip** could be used instead).

- b. Run the following command on the X server to which you are connected:

```
# xhost +
```

3. Run the wizard launcher, `launchpad_aix`, from the AIX directory of the CD by typing the following command:

```
# ./launchpad_aix
```

This will start the CIM Agent LaunchPad, a small program that launches the wizard.

4. Choose from the following options in the LaunchPad window:

Product overview

Offers information about the CIM Agent.

Readme file

Offers any last minute product information that did not make it into this installation guide.

User Guide

Offers instructions on how to install the CIM Agent.

Product Web Information

Offers information from the product Web site.

License Information

Offers information about the license of the CIM Agent.

Install the Product

Starts the CIM Agent installation program.

Exit Exit the CIM Agent installation program.

The LaunchPad window (Figure 3 on page 12) remains open (behind the wizard) during the installation. You can access product information after the installation has started. The LaunchPad returns to the forefront when the installation is complete. You can click **Exit** to close the LaunchPad.

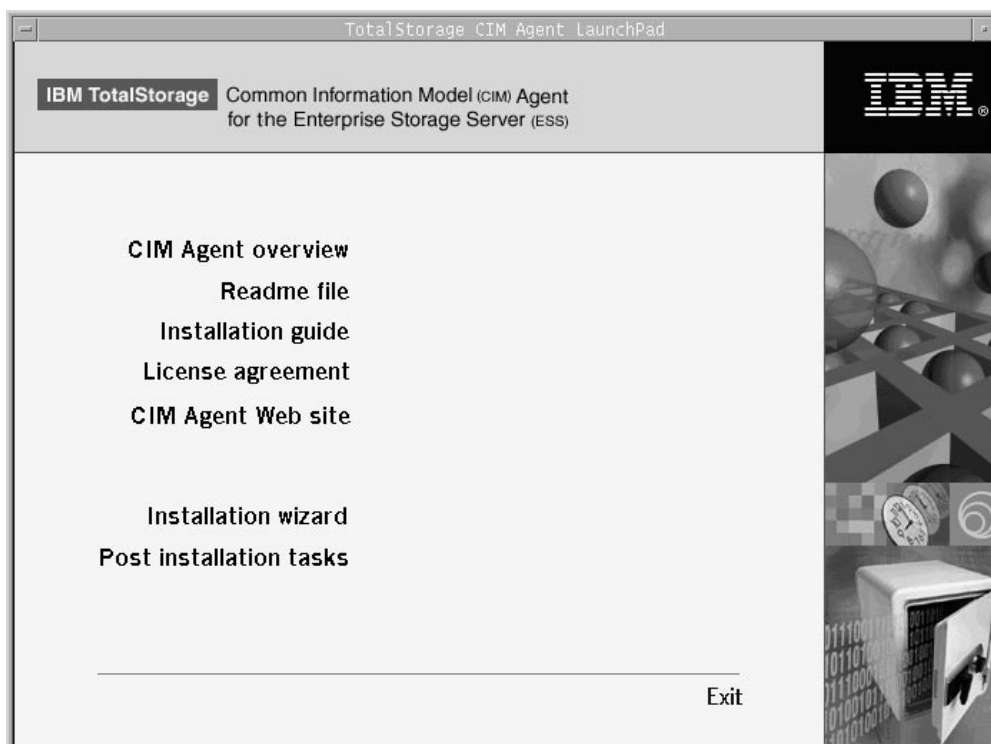


Figure 3. LaunchPad window

5. Check the readme file by clicking the Readme file on the LaunchPad window or by viewing the README.aix file located in the AIX directory on the CIM Agent installation CD. The readme file might provide additional information that supersedes information in this guide.
You can also find this installation guide on the CIM Agent CD under the file name installguide.pdf in the doc subdirectory.
6. Click **Install the Product** to start the installation program.
7. The **Welcome** window opens suggesting which documentation you should review prior to installation. Click **Next** to continue or **Cancel** to exit. See Figure 4 on page 13.

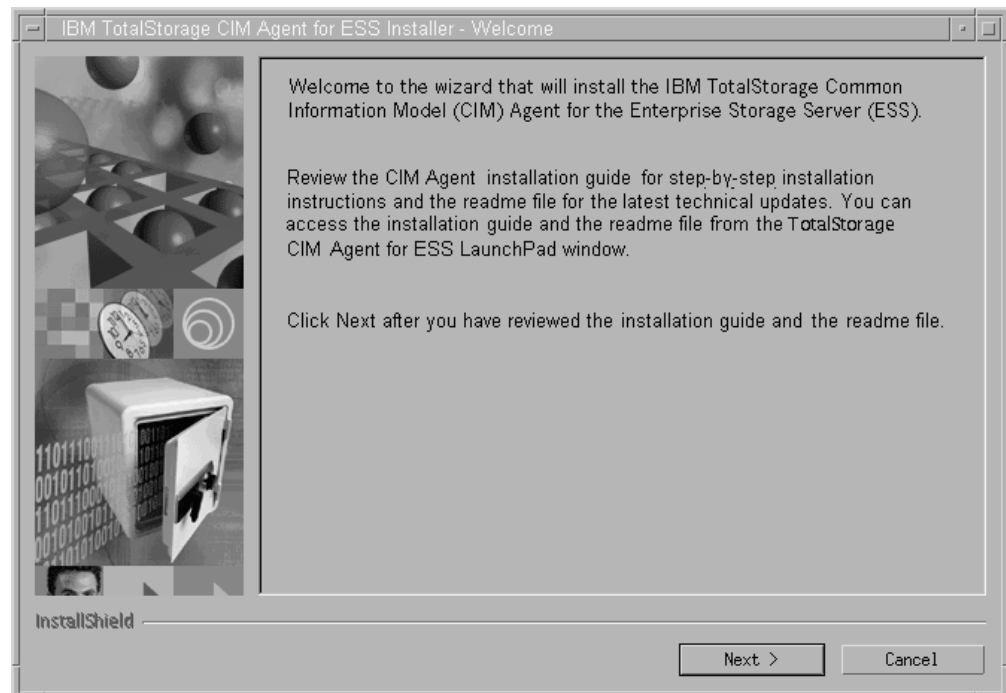


Figure 4. Welcome window

8. Read the license agreement. See Figure 5. Click either **I accept the terms of the license agreement** and click **Next** to proceed, or click **I do not accept the terms of the license agreement** to cancel the installation.

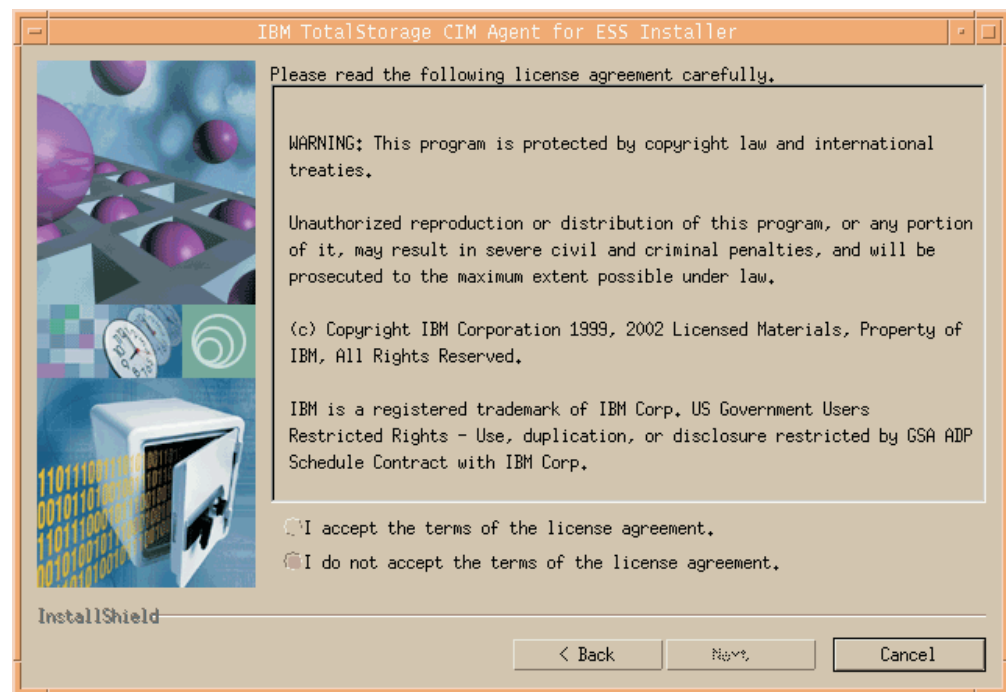


Figure 5. License Agreement window

9. The wizard verifies that your machine meets the installation requirements.

- If you have a service location protocol (SLP) service that is different from the SLP that the CIM Agent requires, the wizard displays an error and asks you to stop the installation and remove this SLP service from the system.
 - The wizard checks if the ESS CLI client is installed on your machine. If the client is installed and its version is 2.1.0.8 or later, the wizard updates the PATH environment variable to include the INSTALL environment variable value.
 - The wizard checks if a version of the CIM Agent is already installed. If the CIM Agent is installed, it checks if the SLP service and the IBM CIM Object Manager (CIMOM) service are started. If these services are started, the program asks if you want to continue the installation program by selecting the **Next** button, or if you want to exit the installation program by selecting the **Cancel** button. If you choose to continue, you must stop all the applications that use these services and you can save the old configuration by selecting the check box that you see on the window.
10. The Select Destination window opens. Click **Next** to accept the default directory, or click **Browse** to select a different directory for installation and then click **Next**. See Figure 6.

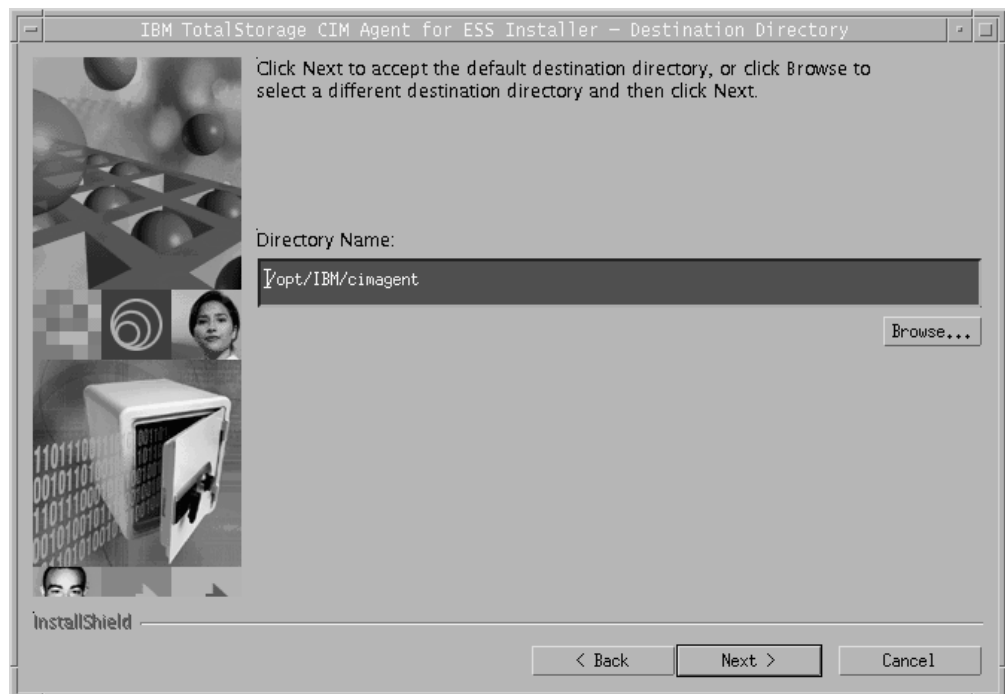


Figure 6. Select Destination window

Note: If the wizard detects insufficient space for the CIM Agent in the chosen directory, an error message is displayed. You must cancel the wizard, free some space in that directory, and restart the wizard. You can also go back by clicking the **Back** button, and choose another directory for the product.

11. The Installation Configuration window opens. Click **Install** to confirm the installation directory and file size, or click **Cancel** to exit the wizard. See Figure 7 on page 15.

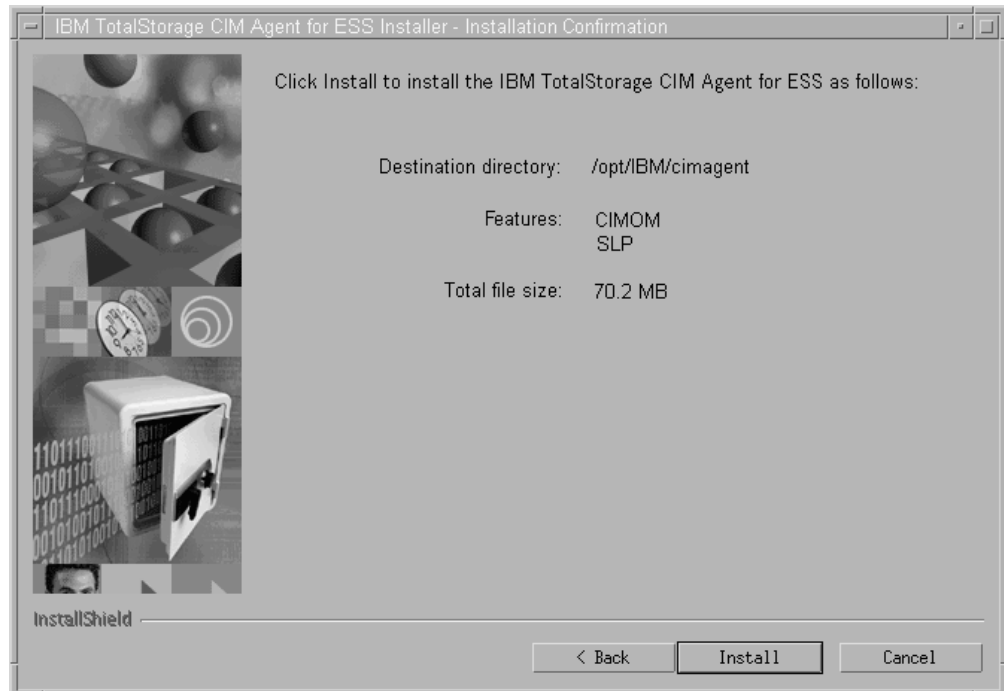


Figure 7. Installation Configuration window

12. The Installation Progress window opens indicating how much of the installation has been completed. See Figure 8. Installation usually takes 3 - 10 minutes depending on your machine configuration. If you do not want to continue with the installation, click **Cancel** to exit.

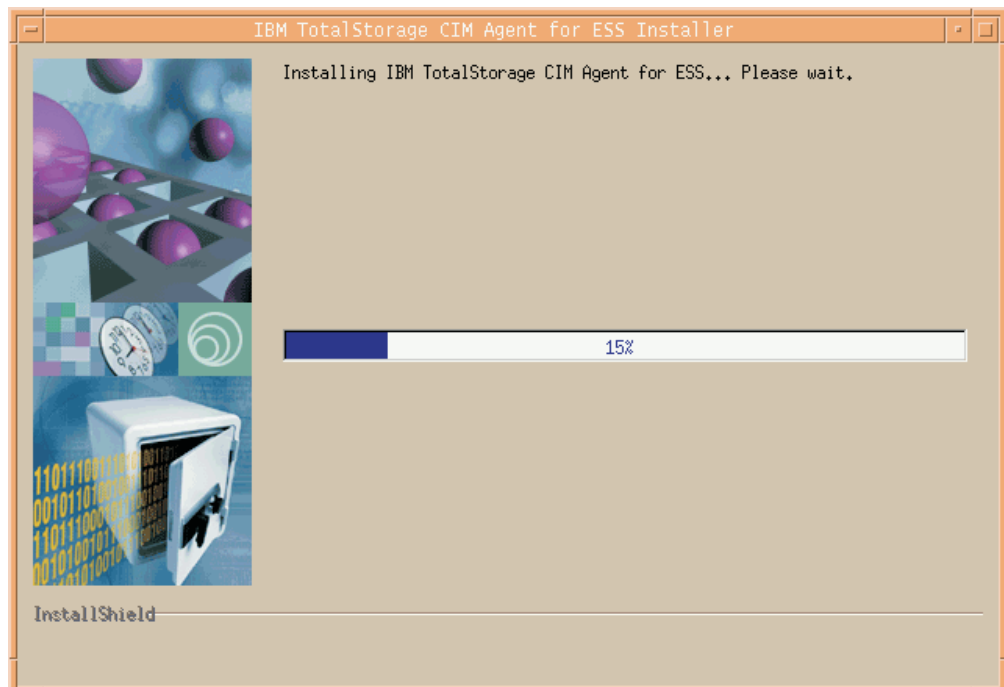


Figure 8. Installation Progress window

13. The Installation Complete window opens displaying if the installation was successful or not. Click **Finish** to exit the wizard. You can view the installation log under `<des-path>/logs/install.log`, where `<des-path>` is your destination directory. See Figure 9.

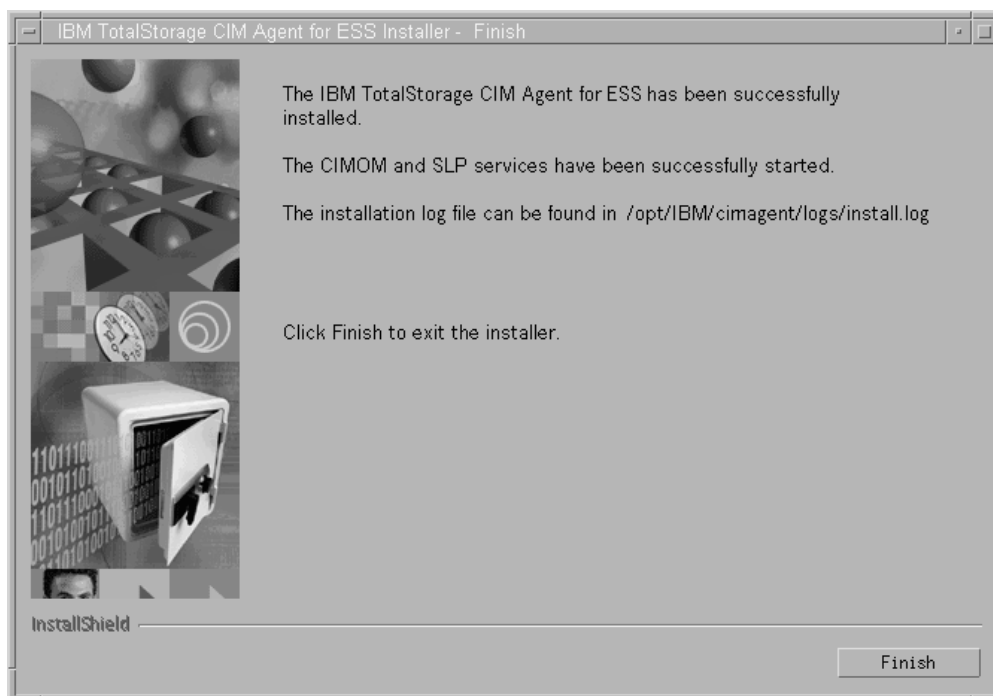


Figure 9. Installation Complete window

14. Click **Exit** to exit the LaunchPad.

Related topics:

- “Verifying the CIM Agent installation for AIX” on page 19
- “Installing the CIM Agent on AIX in unattended (silent) mode”
- “Removing the CIM Agent for AIX” on page 23

Installing the CIM Agent on AIX in unattended (silent) mode

This section includes the steps to install the CIM Agent in your AIX environment using the unattended (silent) mode. You must comply with all prerequisites listed in “CIM Agent installation requirements” on page 4 before you start the installation.

You can choose to install the CIM Agent in unattended (silent) mode, which involves customizing a response file and issuing a command or in graphical mode with the help of an installation wizard. If you want to install the CIM Agent in graphical mode, see “Installing the CIM Agent on AIX in graphical mode” on page 10, otherwise continue with this section. After the completion of either kind of installation, you will then verify the installation of the CIM Agent in section, “Verifying the CIM Agent installation for AIX” on page 19.

Context:

The unattended (silent) installation capability enables you to run an installation process unattended. You can create a standard response file to ensure that the

product is installed consistently on multiple systems. The responsefile file is a template located on the CIM Agent CD-ROM that you must copy to disk and modify. To use the silent mode installation method, you will be performing the following tasks:

1. Find the responsefile file template on the CIM Agent installation compact disk.
2. Copy the responsefile template to your hard drive.
3. Customize the responsefile file to your specifications.
4. Save the updated responsefile file.
5. Invoke the response file using the setupaix script.

Steps:

Perform the following steps to install the CIM Agent in your AIX environment using the silent (unattended) mode:

1. Log on as a user with root authority.
2. Locate the responsefile file on your *Common Information Model Agent* CD.
3. Retrieve and copy the responsefile file to your hard disk drive by typing the following commands:

```
#mkdir /tmp/cimagent
#cp -p /cdrom/AIX/responsefile /tmp/cimagent
```

where:

- */tmp/cimagent* is the directory
- *<response file>* is the path of the responsefile file

You must also modify the responsefile with your desired CIM Agent destination directory (*<dest-path>*).

4. Customize the responsefile file:
 - a. Modify the default parameters in the responsefile file with your desired values in a text editor such as vi:
 - Remove the # character from the beginning of the command line and edit the lines to use values other than the defaults. You **must** enclose all values in double quotation marks ("").
 - The variable *<product.installLocation>* is used to define the default directory where the product will be installed. To use another destination directory, remove the # character from the corresponding line and replace this default directory with the desired directory.
 - The variable *<checkPrerequisite>* enables checking the prerequisites. To disable checking the prerequisites, remove the # character from the corresponding line and change its value to "no".
 - The variable *<startUpgrade>* enables the installation of the product with the same or a higher version. To do this, remove the # character from the corresponding line and change its value to "yes".
 - The variable *<stopProcessesResponse>* tells the program whether or not to stop the SLP and the IBM CIM Object Manager processes if they are active. By default this variable value is "no" and if you do not change this value, the program will abort the installation if these processes are active.
 - To automatically stop the SLP and CIMOM, remove the # character from the corresponding line and change its value to "yes".
 - The variable *<saveConfiguration>* specifies whether to save the configuration files for a later automatically recovering. If you do not want

to save the configuration files, remove the # character from the corresponding line and change its value to "no".

5. Save the modified responsefile file in your desired directory.
6. Type the following command to run the install file:

```
# ./setupaix -options <responsefile-path>/responsefile
```

Note: *<response file path>* is the path of the responsefile file.

7. Wait for the wizard to complete the installation.
8. Check for installation errors in the install.log file. This file can be found in the *<dest-path>/logs* directory. Your install.log file should look similar to the following install.log file:

```
(Nov 8, 2002 4:45:35 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing External Files (SLP Files)
(Nov 8, 2002 4:45:35 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing ICA Set Execute Permissions (Set Execute Permissions1)
(Nov 8, 2002 4:45:35 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing ICA Install SLP (Install Slp)
(Nov 8, 2002 4:45:36 PM), Service Location Protocol successfully installed.
(Nov 8, 2002 4:45:36 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing External Files (IBM JAVA 1.3.1 Files)
(Nov 8, 2002 4:46:09 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing External Files (CIMOM Files)
(Nov 8, 2002 4:46:57 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing ICA Set Execute Permissions (Set Execute Permissions2)
(Nov 8, 2002 4:46:57 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing Bat Replace (Bat replace)
(Nov 8, 2002 4:46:57 PM), The file replacing starts ...
(Nov 8, 2002 4:46:58 PM), The file setupCmdLine successfully replaced
(Nov 8, 2002 4:46:58 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing Compile Mof (Compile Mof)
(Nov 8, 2002 4:46:58 PM), CimAgent, com.ibm.itcins.ical.CompileMof, msg1,
Compile MOF files started ...
(Nov 8, 2002 4:47:22 PM), CimAgent, com.ibm.itcins.ical.CompileMof, msg1,
MOF files successfully compiled.
(Nov 8, 2002 4:47:22 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing Cert Man (Cert Man)
(Nov 8, 2002 4:47:22 PM), CimAgent, com.ibm.itcins.ical.CertMan, msg1,
Generate a certificate store started ...
(Nov 8, 2002 4:47:43 PM), CimAgent, com.ibm.itcins.ical.CertMan, msg1,
Certificate store called truststore successfully generated.
(Nov 8, 2002 4:47:43 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing ICA Install CIM (Install Cimom)
(Nov 8, 2002 4:47:44 PM), CIMOM service successfully installed.
(Nov 8, 2002 4:47:44 PM), CimAgent, com.ibm.wizard.platform.aix.AixProduct
ServiceImpl, msg1, installing AIX Update Inventory Message (Update SWPD)
(Nov 8, 2002 4:47:44 PM), Updating the Software Vital Product Data.
This action will take several minutes. Please wait ...
(Nov 8, 2002 4:49:57 PM), Servers successfully started.

(Nov 8, 2002 4:49:57 PM), The install log file can be found in /opt/IBM/
cimagent/logs/install.log.
You can now configure the product:
- Configure users
- Configure device communications
Return to the Launchpad application if you wish to read product information/
documentation.
```

Note: If the installation fails before the target *<dest-path>* directory is created, you can find the temporary log in */tmp/cimagent/install.log*.

Related topics:

- "Verifying the CIM Agent installation for AIX" on page 19
- "Installing the CIM Agent on AIX in graphical mode" on page 10

- “Configuring the CIM Agent for AIX” on page 20
- “Removing the CIM Agent for AIX” on page 23

Verifying the CIM Agent installation for AIX

This section includes the steps necessary to verify that your CIM Agent is installed correctly on your AIX operating system.

Steps:

Perform the following steps to verify your CIM Agent installation:

1. Verify the installation of the service location protocol (SLP).
 - a. Open a Command Prompt window and type the following command to verify that SLP is started:

```
# ps -ef | grep slpd
```

If the SLP daemon is started, the following output is displayed:

```
root    26438 26986 0 12:36:44 pts/3 0:00 grep slpd
daemon 20730      1 0 12:51:47 -    0:00 /opt/IBM/cimagent/slp/slpd
```

2. Verify the installation of the CIM Agent.
 - a. Check that the CIMOM daemon is installed and started by typing the following command:

```
# ps -ef | grep CIMOM
```

The following is a sample output:

```
# ps -ef |grep cimom
root 32518      1 0 15:52:05 -    0:08 /opt/IBM/cimagent/ibmjava131/jre/
bin/java -Xms128m -Xmx256m -cp /opt/IBM/cimagent/lib/xml4j-4_0_5/xercesImpl.jar:
/opt/IBM/cimagent/lib/xml4j-4_0_5/xmlParserAPIs.jar:/opt/IBM/cimagent/ibmjava131/
jre/lib/ext/ibmjsse.jar:/opt/IBM/cimagent/lib/JCE/IBMJCEfw.jar:/opt/IBM/cimagent/
lib/JCE/IBMJCEProvider.jar:/opt/IBM/cimagent/lib/JCE/local_policy.jar:/opt/IBM/
cimagent/lib/JCE/US_export_policy.jar:/opt/IBM/cimagent/lib/JCE/ibmpkcs.jar:
/opt/IBM/cimagent/ibmjava131/jre/lib/rt.jar:/opt/IBM/cimagent/ibmica.jar
com.ibm.cimom.CIMOM
root 28838 23968 0 15:57:21 pts/0 0:00 grep cimom
```

3. Start the CIMOM, if it is not started, by typing the following command:

```
# <dest-path>/startcimom
```

where *<dest-path>* is the destination directory where the CIM Agent is installed.

4. When you are finished with the CIM Agent CD-ROM, you can release the CD-ROM with the **umount** command, for example,

```
umount /dev/cd0
umount /cdrom
```

Result:

If you are able to perform all of the verification tasks successfully, then the CIM Agent has been successfully installed on your AIX operating system.

Related topics:

- “Configuring the CIM Agent for AIX” on page 20

Configuring the CIM Agent for AIX

This section includes the steps to configure the CIM Agent after it has been successfully installed.

Context:

This task is performed after successful installation of the CIM Agent. This installation program performs the following tasks:

- Copies the directories and files from the ESS CIM Agent CD-ROM into the chosen destination directory
- Updates some script files using the selected destination
- Compiles the MOF files, creates the persistent subdirectory in the destination directory
- Generates a certificate called **truststore** in the destination directory, only for the server

Note: For a client installation, you must copy this certificate from the server and install it in the destination directory on the client machine.

Steps:

Perform the following steps to configure the CIM Agent:

1. Configure the CIMOM for each user who will have authority to use the CIMOM by running the CIMOM configuration program.
 - a. Type the following command:

```
# <dest-path>/setuser
```

where *<dest-path>* is the destination path where the CIM Agent is installed.

The following is a sample output:

```
# ./setuser
Application setuser started in interactive mode
To terminate the application enter: exit
To get a help message enter: help
```

- b. Obtain a user name and password for each user that can manage the CIMOM. Type the following command for each user:

```
>>>adduser <username> <password>
```

The following is a sample output:

```
An entry for user nnnn successfully added
```

where *nnnn* is your userid.

- c. Type the **exit** command to exit the CIMOM configuration program. A file named *cimom.passwd* is created in the destination directory.
2. Configure the CIM Agent for each ESS to which the CIM Agent can have access.
 - a. Type the following command:

```
# <dest-path>/setdevice
```

where *<dest-path>* is the destination directory where the CIM Agent is installed.

The following is a sample output:

```
# ./setdevice
Application setdevice started in interactive mode
To terminate the application enter: exit
To get a help message enter: help
```

- b. Obtain an IP address, user name, and password for each ESS that the CIM Agent is to manage.
- c. Type the following command for each ESS:
 >>>address <IP address> <username> <password>

The following is a sample output:

```
An ess provider entry for IP x.xx.xxx.xx successfully added
```

where x.xx.xxx.xx is the IP address.

- d. Type the **exit** command to exit the CIMOM configuration program. A file named provider-cfg.xml is created in the destination directory.

Result:

If you are able to perform all of the configuring tasks successfully, then the CIM Agent has been successfully installed on your AIX operating system.

Related topics:

- “Running the CIM Agent on AIX” on page 22

Configuring the CIM Agent to run in unsecure mode

Some vendor software might not be capable of communicating with the CIM Agent in a secure fashion. If you wish to, you can still use this vendor software by configuring the CIM Agent to run with only basic user and password security.

Steps:

Perform the following steps to configure in unsecure mode:

1. Type the **stopcimom** command in the destination directory to stop the CIMOM.
2. Find the cimom.properties file in the target directory and edit it with a tool such as vi editor. The following is an example:

```
Set ServerCommunication=HTTP
Set Port=5988
Set DigestAuthentication=False
```

3. Type the **startcimom** command to restart the CIMOM.
4. Verify that the server started on Port 5988 by opening the cimom.log file.

Running the CIM Agent on AIX

This section includes the steps you need to run the CIM Agent on your AIX operating system.

Prerequisites:

Before using the CIM Agent, verify that the ESS CLI software has network connectivity to the ESS by issuing the following command from a Command Prompt window:

```
esscli -u essuser -p esspass -s x.xxx.xxx.xxx list server
```

where:

- *essuser* is an Enterprise Storage Server Specialist user name.
- *esspass* is the Enterprise Storage Server Specialist password for that user name.
- *x.xxx.xxx.xxx* is the IP address of the Enterprise Storage Server.

Note: The ESS CLI does not set the PATH variable. Update the PATH environment variable using the path where the CLI for ESS 2105 is installed. See “Installing the ESS CLI for AIX” on page 7 for more information.

A response similar to the following is displayed:

```
Thu Oct 17 10:19:58 PDT 2002 IBM ESSCLI 2.1.0.8
```

Server	Model	Mfg	WWN	CodeEc	Cache	Nvs	Racks
2105.FCA75	F20	IBM	5005076300C00C4B	1.5.2.43	8GB	384MB	1

Steps:

The installation, verification, and configuration steps need to be completed before you run the CIM Agent.

Perform the following to start and verify the service location protocol (SLP) daemon and the CIMOM.

1. Type the following command to see if the SLP daemon is started:

```
ps -ef | grep slpd
```

If the SLP daemon is not started, continue with the next step. If the SLP daemon is started, go to step 3.

2. Start the SLP daemon, if it is not started, type the following command from a separate window.

```
# /etc/rc.slpd
```

This session will remain active until you stop it. You should keep it running as long as the CIM Agent is running.

3. Type the following command to see if the CIMOM is started:

```
ps -ef | grep CIMOM
```

If the CIMOM is not started, go to step 4. If the CIMOM is started, go to step 5 on page 23.

4. Start the CIMOM by running the startcimom script:

```
# <dest-path>./startcimom
```

Where *<dest-path>* is the destination directory where the CIM Agent is installed. The default is to start the secure CIMOM. It will register itself with SLP and accept requests on Port 5989.

5. Type the following command to locate all WBEM services (for example, CIMOMs) in the local network.

Type the following command from a Command Prompt window:

```
# <dest-path>/verifyconfig -u <user> -p <password>
```

where *<user>* and *<password>* are the userid and passwords for a CIMOM user that was created using **setuser** command.

A message similar to the following is displayed:

```
# ./verifyconfig -u xxxx -p nnnn
Verifying configuration of ESS CIM Agent...
Communicating with SLP to find WBEM services...
3 WBEM services found
  host=baboon, port=5988
  host=gorilla, port=5988
  host=tpc03, port=5988
Connecting to ESS CIM Agent, host=baboon, port=5988
Found 1 IBMTSESS_StorageSystem instances
Verification Successful
```

where *xxxx* is your userid and *nnnn* is your password.

Result:

The CIM Agent is now running.

Removing the CIM Agent for AIX

This optional task includes the steps necessary to remove the CIM Agent from your AIX operating system.

Steps:

Perform the following steps to remove the CIM Agent:

1. Log on as a user with root authority.
2. Stop all the processes and applications that use SLP and CIMOM, that are running on your system before you remove the CIM Agent for ESS.
3. Type the following command to stop the IBM CIM Object Manager (CIMOM) daemon.

```
# <dest-path>/stopcimom
```

where *<dest-path>* is the destination directory where the product was installed.

4. Type the following command to see if SLP daemon is running:

```
# ps -ef | grep slpd
```

If SLP is running, the following output is displayed:

```
# ps -ef |grep slpd
daemon 24452      1   0   Nov 13   -   0:00 /opt/IBM/cimagent/slp/slpd
root 37756 27328   0 14:55:47 pts/1   0:00 grep slpd
```

5. Type the following command to stop the service location protocol (SLP):

```
# kill 24452
```

6. Run the removal program in graphical mode (see “Removing in graphical mode”) or in unattended (silent) mode (see “Removing in unattended (silent) mode” on page 27) to remove the IBM CIM Object Manager Service and Service Location Protocol.

Removing in graphical mode

Perform the following steps to remove the CIM Agent using graphical mode:

1. Type the following command to run the removal program from the `_uninst` subdirectory of the `<dest-path>`:

```
# <dest-path>/_uninst/uninstaller
```

where `<dest-path>` is the destination directory where the CIM Agent is installed.

2. The removal program was not created during the CIM Agent installation, type the following command:

```
# <dest-path>/ibmjava131/jre/bin/java -jar <dest-path>/_uninst/uninstall.jar
```

where `<dest-path>` is the destination directory where the CIM Agent is installed.

3. The Welcome window opens. See Figure 10. Click **Next** to continue with the removal program, or click **Cancel** to exit the removal program.

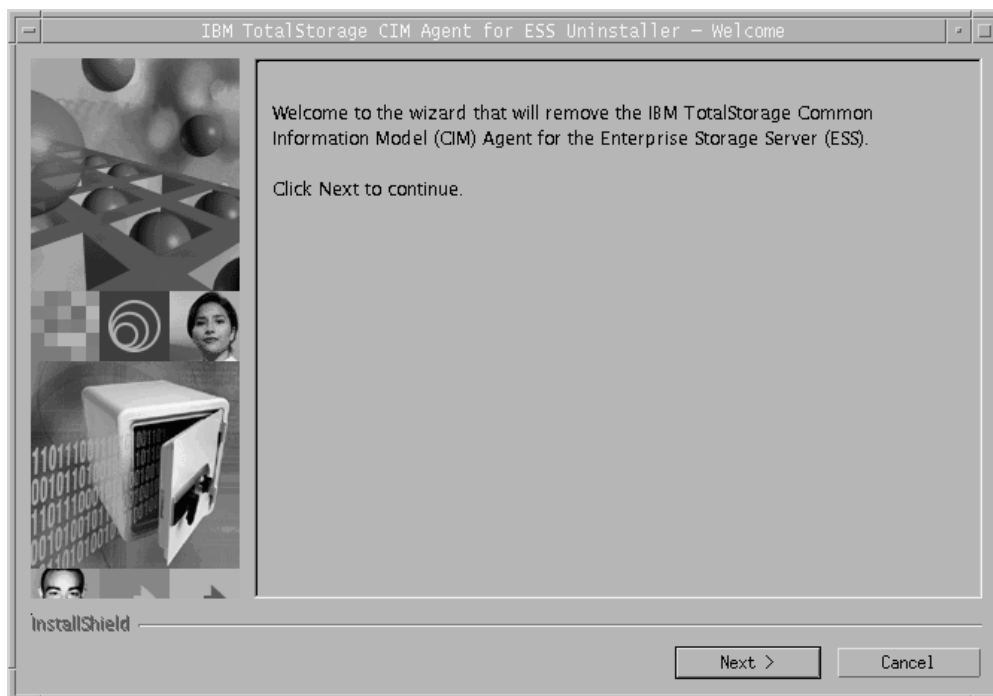


Figure 10. Welcome window

The program detects if the service location protocol (SLP) and the IBM CIM Object Manager (CIMOM) services are running and displays the following information:

- If they are running, the program asks if you want to continue with the removal program. In that case, if you click **Next** the program will stop the services, or if you click **Cancel** you will have to stop the services yourself (manually).
- If you want to manually stop the services, you must exit the removal program, stop the services and the applications that use them and then run the removal program again from the beginning.

Attention: You must be careful if you have other applications that use the SLP service. If you do, you must stop these applications before stopping the SLP service because the SLP service will be deleted during the removal process. See Figure 11.

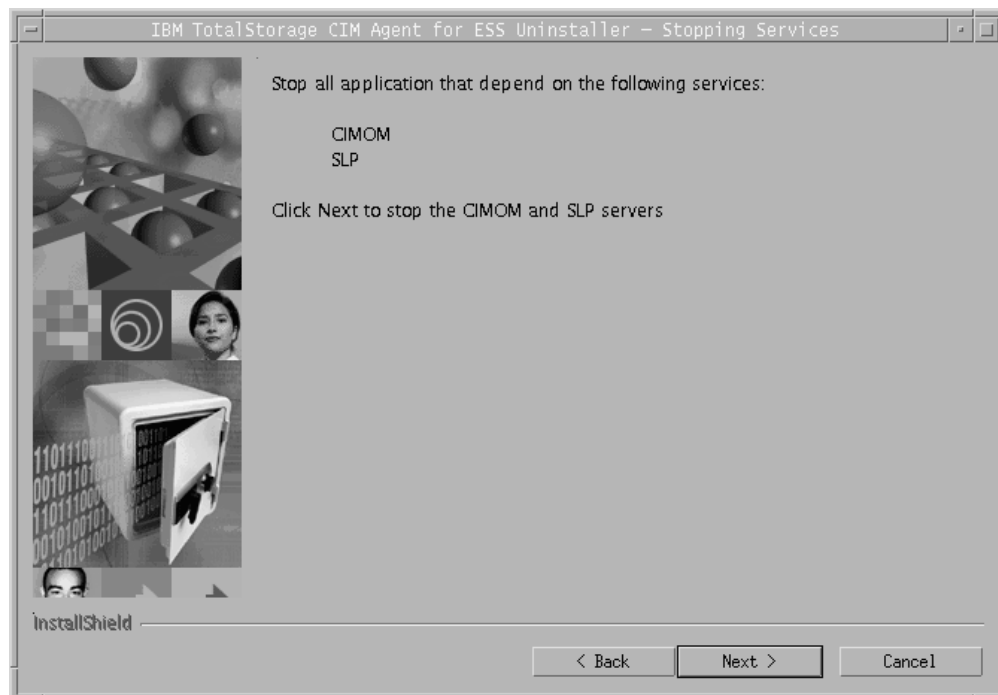


Figure 11. Stop SLP and CIMOM window

- If neither the SLP service nor the CIMOM service is running, the removal program continues with the Preview window.
4. The Preview window opens, displaying the location of the product that will be removed. See Figure 12 on page 26. Click **Remove** to continue with the removal program, or click **Cancel** to exit.



Figure 12. Preview window

5. Wait for the program to remove CIM Agent.
6. The Uninstallation Complete window opens displaying information about the result of removal (successfully or failed). See Figure 13.

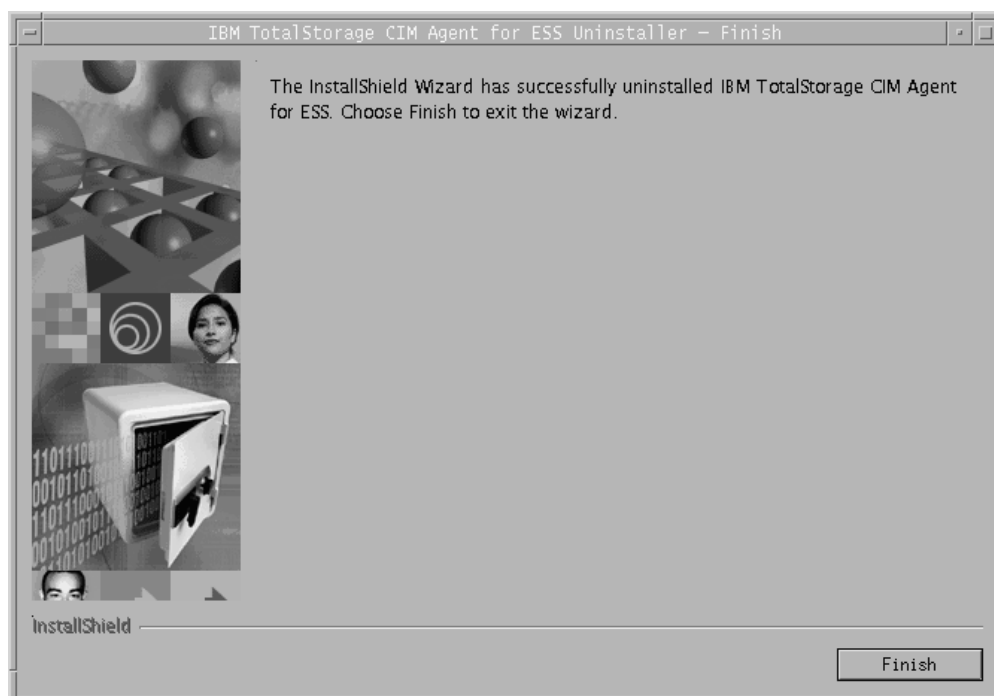


Figure 13. Summary window

Click **Finish** to end the removal program.

Removing in unattended (silent) mode

Steps:

Perform the following steps to remove the CIM Agent in unattended (silent) mode:

1. Stop SLP, CIMOM, and all related processes by typing the following command:

```
<dest-path>/_uninst/uninstaller -silent -G stopProcessesResponse=yes
```

2. Type the following command to start the removal process:

```
<dest-path>/_uninst/uninstaller -silent
```

Post-processing requirements:

The CIM Agent removal process does not remove configuration files, logs, and similar files that are created during or after the installation process. They are located in the destination path where CIM Agent was installed. For example, the default destination path is /opt/IBM/cimagent.

Remove the directory and all of its contents (especially if you plan to reinstall the CIM Agent).

Note: If you want to keep the old configuration files, save them in another location on your system before removing them from the installation destination path, so you can restore them later.

To remove the directory, cimagent, you must type the following command, for example, from the IBM directory.

```
# rm -r /opt/IBM/cimagent
```

Note: The recursive remove is used in the following example because the CIM Agent has a deep directory structure. The recursive remove is very powerful and dangerous. You should use the fully qualified directory name.

Chapter 3. CIM Agent for Linux

This chapter includes an overview of the installation process and instructions for installing and configuring the CIM Agent on a Linux operating system.

Installation overview for Linux

This section provides an overview of the installation and configuration of the CIM Agent on a Linux operating system. You should have some knowledge of how to a Linux operating system before you install the CIM Agent. You should also become familiar with the command that you use during installation and configuration of the CIM Agent. See Chapter 5, “CIM Agent commands”, on page 73 for information about the commands.

The following list of installation and configuration tasks are in the order in which they should be performed:

1. Before you install the CIM Agent on Linux, you should check the hardware and software requirements listed in “CIM Agent installation requirements” on page 4.
2. Install the prerequisite ESS CLI software, as instructed in “Installing the ESS CLI for Linux”.
3. You can choose to install the CIM Agent either in graphical mode with the help of an installation wizard or in unattended mode (also known as silent mode), which involves customizing a response file and issuing a command.
 - a. If you want to install the CIM Agent in graphical mode perform the instructions in “Installing the CIM Agent on Linux in graphical mode” on page 31.
 - b. If you want to install the CIM Agent in unattended mode perform the instructions in “Installing the CIM Agent on Linux in unattended (silent) mode” on page 38.
4. Configure the CIM agent for Linux by performing the instructions in “Configuring the CIM Agent for Linux” on page 42. You might want to revisit this section in the future as you add, change, or delete CIMOM authentication and ESS information.
5. Verify connection to your ESS by performing the instructions in “Running the CIM Agent on Linux” on page 44.
6. Instructions for removing the CIM Agent are also provided in “Removing the CIM Agent for Linux” on page 46. You only need to perform this optional task if you get errors during installation verification or if the CIM Agent did not set the environment variables.

Installing the ESS CLI for Linux

The CIM Agent has a software prerequisite of the IBM TotalStorage Enterprise Storage System Command Line Interface (ESS CLI).

Steps:

Perform the following steps to install the ESS CLI for Linux:

1. Install the ESS CLI before you install CIM Agent.
This software must be installed before you install the CIM Agent. The ESS CLI must be installed first because the CIM Agent will set the path information in batch commands for you based on the ESS CLI location on the disk. The CIM

Agent installation program will check your system for the existence of the ESS CLI and will not continue if the ESS CLI is not installed.

2. Select the *ESS CLI* CD to install the ESS CLI.

Note: The CIM Agent supports ESS CLI level 2.1.0.8 or later.

3. Select the correct software from the *ESS CLI* CD, as it contains two products with similar names.

The product you need can be found in the RedHat subdirectory of the *ESS CLI* CD in the rpm file which has a name beginning with *IBMesscli*. The rpm file including the product found in the RedHat subdirectory which has a name beginning with *IBMcli* is **not** the required prerequisite for the CIM Agent.

4. Let the ESS CLI installation set the system path for you. Respond **yes** when you are prompted by "CLI for ESS 2105" installation program to have it set the system path for you.
5. Determine if the ESS CLI is installed.

You can determine if the ESS CLI is installed by reviewing the installed programs on your RedHat Linux system. The level of the software should be 2.1.0.8 or higher. To check the VRMF-X version, issue the following command:

```
rpm -qa | grep IBMesscli
```

If the ESS CLI is installed, a message similar to the following should be displayed.

```
IBMesscli-V.R.M.F-X
```

Where

- *V* represents the CLI version number
 - *R* represents the release number
 - *M* represents the number of times that the release package was built
 - *F* represents the fix level
 - *X* represents the number of times that the release package was built
6. Verify if the ESS CLI is operational and all the environment variables have been set.

Note: In order to run the *esscli*, you must know where the ESS CLI is installed. You can find out where it is installed by issuing the following command:

```
rpm -ql IBMesscli-2.1.0.8-0 | grep esscli
```

where *2.1.0.8-0* is the VRMF-X version.

To verify if the ESS CLI is operational, issue the following command:

```
# /opt/ibm2105cli/esscli
```

The following is an example output from the previous command:

```
# /opt/ibm2105cli/esscli

Thu Oct 17 10:15:50 PDT 2002 IBM ESSCLI 2.1.0.0

esscli -a AccessFile | -u Username -p Password
-s ServerIpAddr [-b BackupServer]
[-v] [-fmt "Format"] [-nohdr] [-noss1]
list Log
list Task
list Server
list DiskGroup -d "ess=EssId"
list PAV -d "ess=EssId volume=VolId | lss=LssId"
list Port -d "ess=EssId"
list Volume -d "ess=EssId [lss=LssId] [voltype=VolType]"
list VolumeSpace -d "ess=EssId"
list HostConnection -d "ess=EssId"
list PPRCPaths -d "ess=EssId [srcLss=LssId]"
list VolumeAccess [-d "init=WwName | host=HostName |
    ess=EssId volume=ALL | volume=UNASSIGNED | volume=VolId1,VolId2,..."]
create HostConnection -d "init=WwName ess=EssId host=HostName
    profile=ProfileName [ports=PortId1,PortId2,...]"
create Volume -d "ess=EssId [voltype=VolType] [qty=Quantity] cap=Gigabytes"
    [cyls=Cylinders lss=LssId1,LssId2,... | vs=VSId1,VSId2,..."
    [redundancy=RAID5|RAID10] [placement=SEQ|SPREAD]"
create VolumeAccess -d "init=WwName ess=EssId volume=VolId1,VolId2,..."
delete HostConnection -d "init=WwName ess=EssId"
delete VolumeAccess -d "init=WwName ess=EssId
    volume=ALL | volume=VolId1,VolId2,..."
set Volume -d "ess=EssId volume=VolId label=VolLabel"
set Port -d "ess=EssId port=PortId topology=FCAL|P2P|UNDEFINED"
set HostConnection -d "init=WwName ess=EssId newinit=WwName"
set HostConnection -d "init=WwName ess=EssId host=NickName"
set HostConnection -d "init=WwName ess=EssId ports=All | ports=PortId1,PortId2,
    ."
show Task -d "name=TaskName"
#
```

Installing the CIM Agent on Linux in graphical mode

This task will assist you with the installation of the CIM Agent in your Linux environment. You must satisfy all prerequisite requirements listed in “CIM Agent installation requirements” on page 4 before starting the installation.

You can choose to install the CIM Agent in graphical mode with the help of an installation wizard or in silent (unattended) mode, which involves customizing a response file and issuing a command. If you want to install the CIM Agent in silent mode, see “Installing the CIM Agent on Linux in unattended (silent) mode” on page 38, otherwise continue with this section. After the completion of either kind of installation, you will then verify the installation of the CIM Agent in section, “Installing the CIM Agent on Linux in unattended (silent) mode” on page 38. Before you install the CIM Agent on Linux, you should check the hardware and software requirements listed in “CIM Agent installation requirements” on page 4.

Steps:

You should have some knowledge of how to administer a Linux operating system before beginning the installation of the CIM Agent.

Note: The description of commands in this task have the convention of optional and substitution parameters between greater than ">" and less than "<" symbols. You should become familiar with the explanation before entering the command.

Perform the following steps to install the CIM Agent on your Linux system:

1. Log on as a user with root authority.
2. Insert the *Common Information Model Agent* CD.
3. Create a mount point or choose an existing mount point.

To create a mount point called "/mnt/cdrom", type the following command:

```
# mkdir /mnt/cdrom
```

4. To mount the CD-ROM file system at the desired mount point, issue the following command:

```
# mount /mnt/cdrom
```

5. Change the current directory to the mount point for the CD-ROM drive, in the LINUX directory. For example, if the CD-ROM was mounted at the /mnt/cdrom, type the following command:

```
# cd /mnt/cdrom/LINUX
```

6. Check the **README.linux** file located in the LINUX directory on the CIM proxy API installation compact disk. The **README.linux** file can provide additional information that supersedes information in this guide.

You can also find this installation guide on the CIM Agent compact disc under the file name **installguide.pdf** in the document subdirectory.

7. Run the installation program launcher, **launchpad_linux** from the Linux directory of the CD-ROM by typing the following command:

```
# ./launchpad_linux
```

This will start the CIM Agent LaunchPad, a small graphic program that launches the installation setup program.

8. The **LaunchPad** window displays, which allows you to choose from the following options:
 - Product Overview - offers information about the CIM Agent
 - ReadMe File - offers any last minute product information that did not make it into the installation guide
 - User Guide - offers instructions on how to install the CIM Agent
 - Product Web Information - offers product technical support
 - License Information - offers information about the CIM Agent license
 - Install the product - starts the CIM Agent installation
 - Exit - exits the CIM Agent installation

The **LaunchPad** window remains open (behind the installer) during the installation. This provides you with access to product information after the installation has started. The LaunchPad returns to the forefront when the installation is complete, you can click **Exit** to close the LaunchPad. See Figure 14 on page 33.

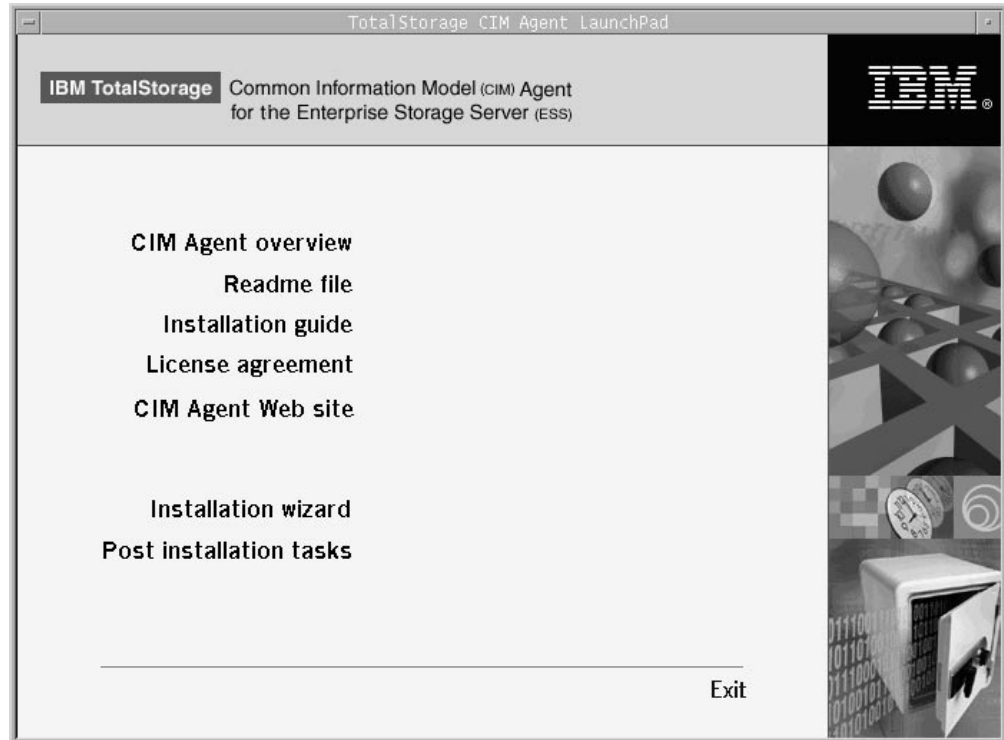


Figure 14. LaunchPad window

9. Select **Install the Product** to begin the CIM Agent installation.
10. The **Welcome** window appears suggesting which documentation you should review prior to installation. Click **Next** to continue. See Figure 15 on page 34.

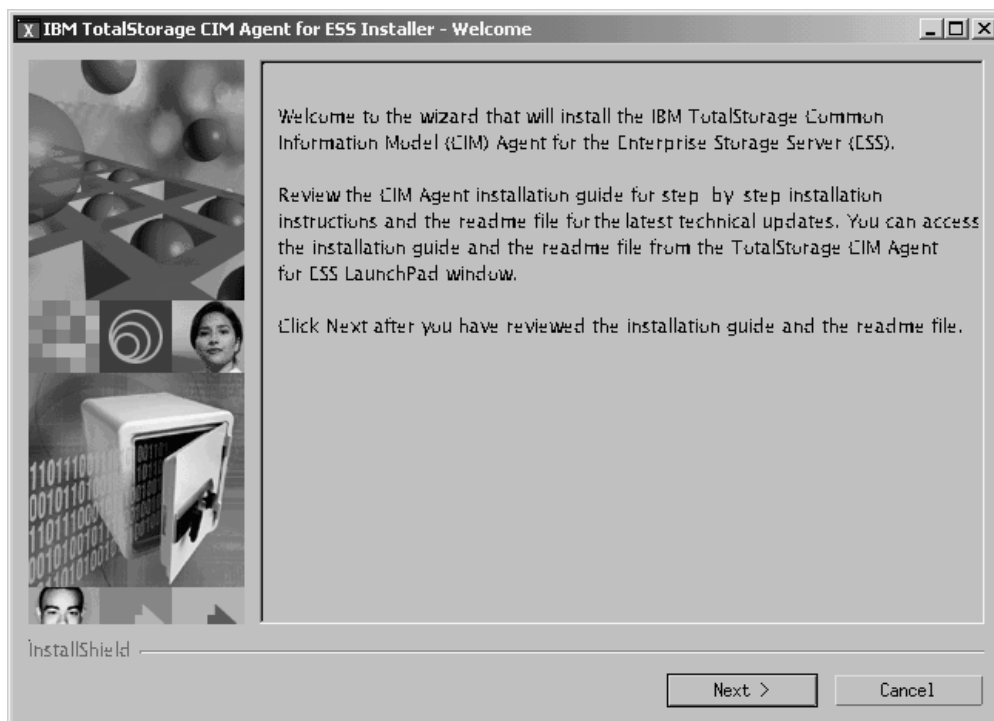


Figure 15. Welcome window

11. Read the license agreement information. Click **I accept the terms of the license agreement** if you accept the license agreement and then click **Next**. Otherwise, click **I do not accept the terms of the license agreement** and then click **Cancel** to exit the installation. See Figure 16 on page 35.

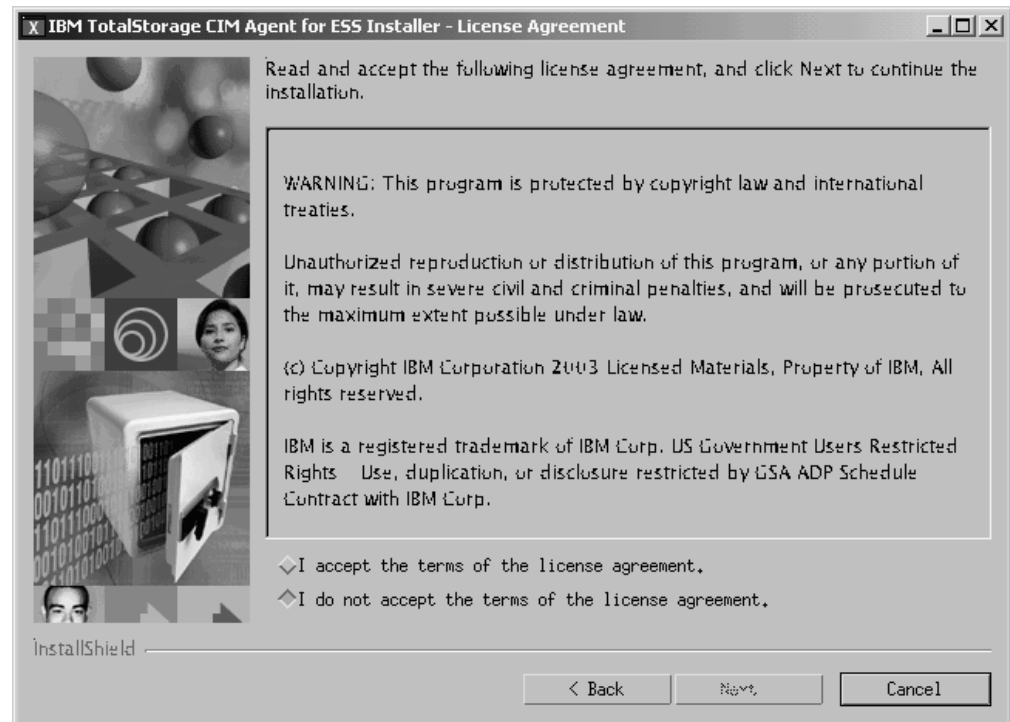
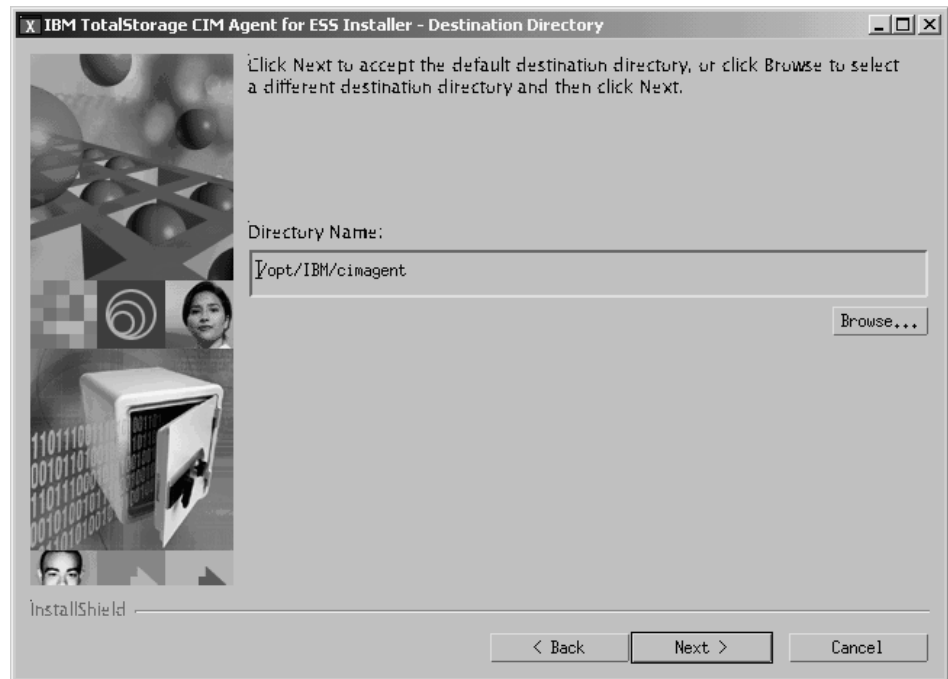


Figure 16. License Agreement window

12. The installation program performs some verifications in your system, displaying the following information:
 - If a Service Location Protocol service is different from the SLP installed by the CIM Agent installer, the program displays an error and asks you to stop and remove this SLP service from the system.
 - The installation program checks if the ESS CLI Client 2105 is installed on your machine. Refer to the *IBM TotalStorage Enterprise Storage Server™: Command-Line Interfaces User's Guide*, SC26-7494-00, for installation information.
 - The installation program checks if a version of the CIM Agent is already installed. If the CIM Agent is installed, it checks if the Service Location Protocol service and the IBM CIM Object Manager service are started. The program asks if you want to continue the installation program by selecting the **Next** button, (in this case, the program will stop the daemons if they are running) or if you want to exit the installation program by selecting the **Cancel** button (in this case, you must manually stop the SLP and CIMOM daemons and the applications that use them and then restart the installation). If you choose to continue, you must stop all the applications that use these services and you can save the old configuration by selecting the check box that you see on the screen.

13. The **Select Destination** window opens. Click **Next** to accept the default directory, or **Browse** to select a different directory for installation. See Figure 17.

Figure 17. Select Destination window



Note: If the program detects insufficient space for the CIM Agent installation in the chosen destination, an error message appears. You must cancel the installation program, free some space in that destination, and restart the installation program. You can also go back by clicking the **Back** button, and choose another destination for the product.

14. Click **Install** to confirm the installation directory and file size. See Figure 18 on page 37.

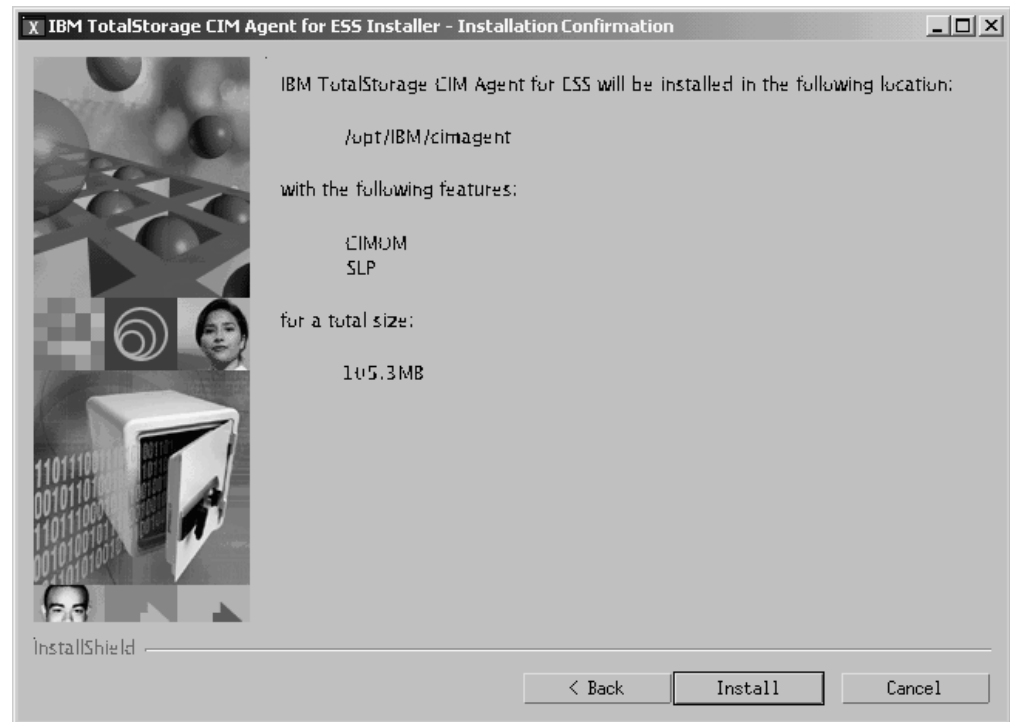


Figure 18. Installation Preview window

15. The **Installation Progress** screen appears indicating how much of the installation has been completed. Installation usually takes 3 - 10 minutes depending on your machine configuration. See Figure 19.

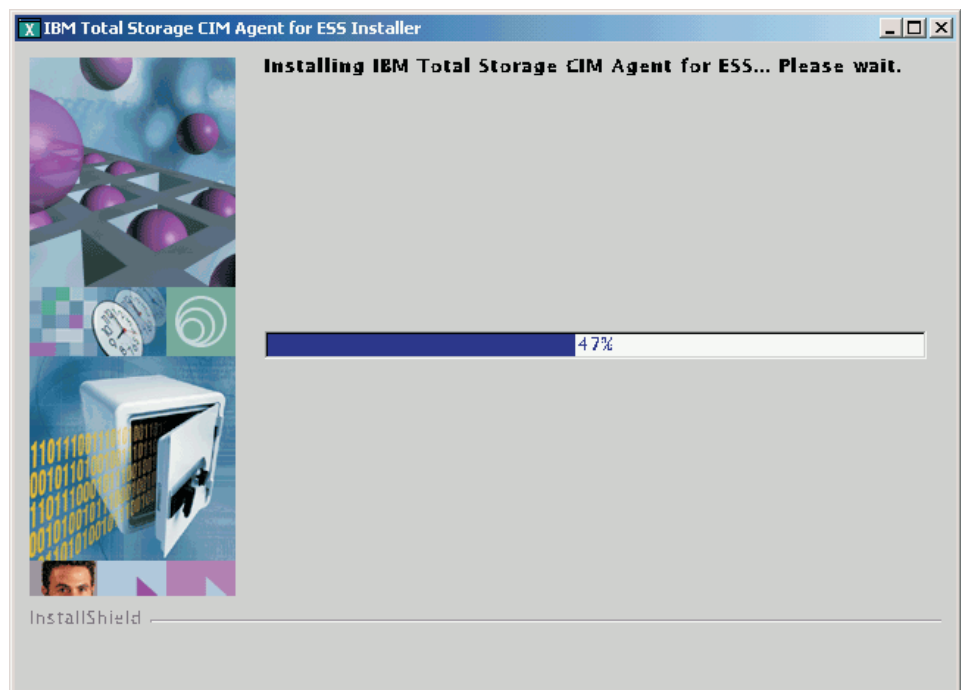


Figure 19. Installation Progress window

16. The **Installation Complete** window appears, displaying the result of the installation. Click **Finish** to complete the setup procedures. See Figure 20.

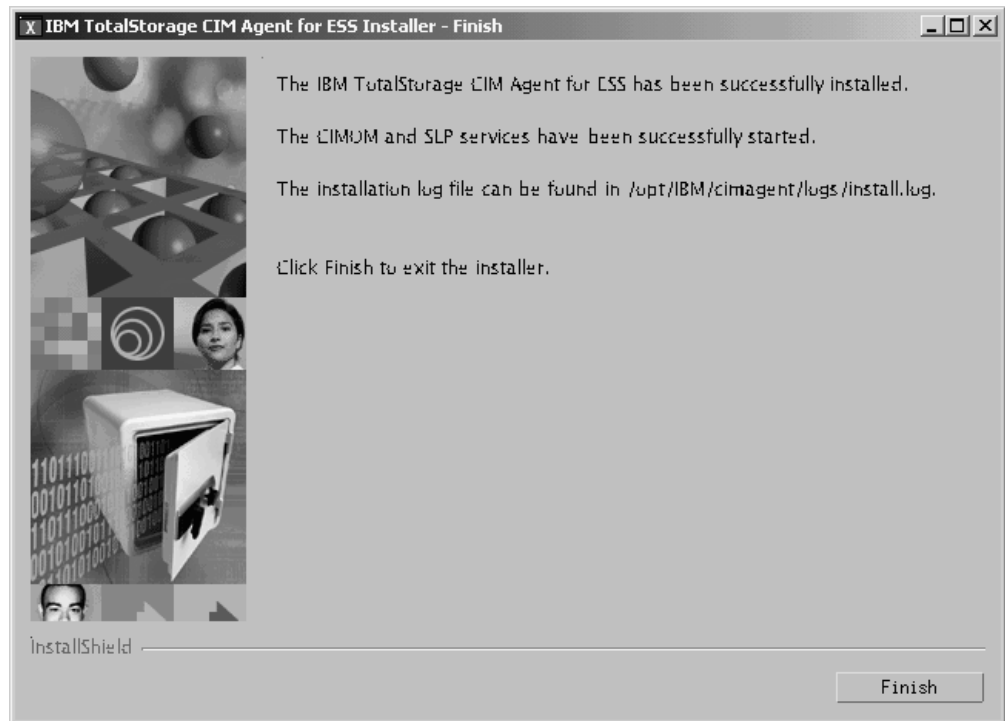


Figure 20. Installation Complete window

17. Exit the LaunchPad program by clicking the **Exit** button on the **LaunchPad** window.
18. When you are finished with the CIM Agent CD-ROM, you can issue the following command to remove the CD-ROM drive:

```
# umount/mnt/cdrom
```

Where /mnt/cdrom is where the CD-ROM is mounted.

Related topics:

- “Installing the CIM Agent on Linux in unattended (silent) mode”
- “Verifying the CIM Agent installation for Linux” on page 42
- “Installing the CIM Agent on Linux in graphical mode” on page 31
- “Removing the CIM Agent for Linux” on page 46

Installing the CIM Agent on Linux in unattended (silent) mode

This task will assist you with the installation of the CIM Agent API in your Linux environment using the unattended (silent) mode. This method of installation involves customizing a response file and issuing a command from a Command Prompt window. You must satisfy all prerequisite requirements before starting the installation. See “CIM Agent installation requirements” on page 4 for prerequisite information.

Context:

The silent install capability enables you to run an installation and uninstallation unattended. You can create a standard response file to ensure the product is installed consistently on multiple systems. The **responsefile** file is a template located on the CIM Agent CD-ROM that you must copy to disk and modify. To use the silent mode installation method, you will be performing the following tasks:

1. Find the **responsefile** file template on the *Common Information Model Agent* CD.
2. Copy the responsefile template to your hard drive.
3. Customize the file to your specifications.
4. Save the updated **responsefile** file.
5. Invoke the response file using the **setuplinux** script.

Steps:

Perform the following steps to install the CIM Agent in your Linux environment using the silent mode:

1. Log on as a user with root authority.
2. Locate the **responsefile** file on your *Common Information Model Agent* CD by typing the following commands:

```
cd /mnt/cdrom
cd LINUX
```

3. Copy the **responsefile** file to your hard drive by typing the following command:

```
cp ./responsefile /root/xxxx
```

where xxxx is your desired directory.

4. Customize the **responsefile** file, with your parameters. The **responsefile** file is a template located on the CIM Agent compact disk that you must modify, as follows:
 - a. Read the directions and copy the **responsefile** from the CD-ROM to a location on your hard disk.
 - b. Modify the default parameters in the **responsefile** file with your desired values in a text editor:
 - Remove the # character from the beginning of the command line and edit the lines to use values other than the defaults. You **must** enclose all values in double quotation marks ("").
 - The option *-P product.installLocation* is used to define the location where the product will be installed. Remove the # character from the corresponding line and replace the default directory with the desired directory.
 - The variable *<checkPrerequisite>* enables checking the prerequisites. To disable checking the prerequisites, remove the # character from the corresponding line and change its value to "no".
 - The variable *<startUpgrade>* enables the installation of the product with the same or a higher version. To do this, remove the # character from the corresponding line and change its value to "yes".
 - The variable *<stopProcessesResponse>* tells the program whether or not to stop the SLP and the IBM CIM Object Manager processes if they are active. By default this variable value is "no" and if you do not change this value, the program will abort the installation if these processes are active.

- To automatically stop the SLP and CIMOM, remove the # character from the corresponding line and change its value to "yes".
 - The variable `<saveConfiguration>` specifies whether to save the configuration files for a later automatically recovering. If you do not want to save the configuration files, remove the # character from the corresponding line and change its value to "no".
5. Save the modified responsefile in your desired directory.
 6. To launch the installer in silent (unattended) mode with the customized responsefile, run the following command from the Linux directory on your CIM Agent CD-ROM:

```
# ./setuplinux -options <responsefile-path>/responsefile
```

Note: `<response file path>` is the path of the responsefile file.

7. Wait for the installation program to complete the installation.
8. Check for installation errors in the install.log file. The log file is initially created in `/tmp/cimagent/install.log`. At the end of the installation, the log can be found in `<dest-path>/logs/install.log`, where `<dest-path>` is the destination directory where the CIM Agent was installed. If the installation ends before the creation of `<dest-path>` the log should be looked for in `/tmp/cimagent/install.log`. Your install.log file should look similar to the following:


```

(Nov 8, 2002 4:33:02 PM), Detected following package installed.
IBMesscli 2.1.0.8 0
(Nov 8, 2002 4:33:02 PM), Command to be executed:
/sbin/ldconfig -p | grep libslp.so
(Nov 8, 2002 4:34:38 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing External Files (SLP Files)
(Nov 8, 2002 4:34:38 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing ICA Set Execute Permissions (Set Execute
Permissions1)
(Nov 8, 2002 4:34:38 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing ICA Install SLP (Install Slp)
(Nov 8, 2002 4:34:38 PM), Copy file /opt/IBM/cimagent/slp/slp.conf to /etc/slp.
conf .
(Nov 8, 2002 4:34:38 PM), Command to be executed:
/bin/ln -sf libslp.so.1.0.0 /opt/IBM/cimagent/slp/libslp.so
(Nov 8, 2002 4:34:38 PM), Command to be executed:
/bin/ln -sf libslp.so.1.0.0 /opt/IBM/cimagent/slp/libslp.so.1
(Nov 8, 2002 4:34:38 PM), Copy file /mnt/cdrom/LINUX/startup/slpdREDHAT to /etc/
init.d/slpd .
(Nov 8, 2002 4:34:38 PM), Command to be executed:/sbin/chkconfig --add slpd
(Nov 8, 2002 4:34:38 PM), Service Location Protocol successfully installed .
(Nov 8, 2002 4:34:39 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing External Files (IBM JAVA 1.3.1 Files)
(Nov 8, 2002 4:34:58 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing External Files (CIMOM Files)
(Nov 8, 2002 4:35:21 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing ICA Set Execute Permissions (Set Execute Permissions2)
(Nov 8, 2002 4:35:21 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing Bat Replace (Bat replace)
(Nov 8, 2002 4:35:21 PM), The file replacing starts ...
(Nov 8, 2002 4:35:21 PM), The file setupCmdLine successfully replaced
(Nov 8, 2002 4:35:21 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing Compile Mof (Compile Mof)
(Nov 8, 2002 4:35:21 PM), CimAgent, com.ibm.itcins.ical.CompileMof, msg1,
Compile MOF files started ...
(Nov 8, 2002 4:35:56 PM), CimAgent, com.ibm.itcins.ical.CompileMof, msg1,
MOF files successfully compiled.
(Nov 8, 2002 4:35:56 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing Cert Man (Cert Man)
(Nov 8, 2002 4:35:56 PM), CimAgent, com.ibm.itcins.ical.CertMan, msg1, Generate
a certificate store started ...
(Nov 8, 2002 4:36:16 PM), CimAgent, com.ibm.itcins.ical.CertMan, msg1, Certificate
store called truststore successfully generated.
(Nov 8, 2002 4:36:16 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, msg1, installing ICA Install CIM (Install Cimom)
(Nov 8, 2002 4:36:16 PM), Copy file /mnt/cdrom/LINUX/startup/cimomREDHAT to /etc/
init.d/cimom .
(Nov 8, 2002 4:36:16 PM), Command to be executed:
/sbin/chkconfig --add cimom
(Nov 8, 2002 4:36:16 PM), CIMOM service successfully installed .
(Nov 8, 2002 4:36:25 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, wrn, The RPM package ibmcimagent.slp-1.0-0 could not be entered
into RPM Db
(Nov 8, 2002 4:36:32 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, wrn, The RPM package ibmcimagent.cimom-1.0-0 could not be entered
into RPM Db
(Nov 8, 2002 4:36:32 PM), CimAgent, com.ibm.wizard.platform.linux.LinuxProduct
ServiceImpl, wrn, The install could not successfully add the product information
into the RPM database. Installation will continue as this is not critical to
the installation of the product.
(Nov 8, 2002 4:37:01 PM), Servers successfully started.

(Nov 8, 2002 4:37:01 PM), The install log file can be found in /opt/IBM/cimagent/
logs/install.log.
You can now configure the product:
    - Configure users
    - Configure device communications
Return to the Launchpad application if you wish to read product information/
documentation.

```

Related topics:

- “Verifying the CIM Agent installation for Linux”
- “Installing the CIM Agent on Linux in graphical mode” on page 31
- “Configuring the CIM Agent for Linux”
- “Removing the CIM Agent for Linux” on page 46

Verifying the CIM Agent installation for Linux

This task describes the steps necessary to verify that your CIM Agent is installed correctly on your Linux system.

Steps:

Perform the following steps to verify your CIM Agent API installation:

1. Verify the installation of the Service Location Protocol (SLP).
 - a. Open an Command Prompt window and type the following command to verify that SLP is installed:

```
# ps -ef | grep -v grep | grep slpd
```

If the SLP daemon is started, you will see the following output:

```
daemon 16054      1 0 18:54 ?        00:00:00 /opt/IBM/cimagent/slp/slpd
```
2. Verify the installation of TotalStorage CIM Agent Tool.
 - a. Check that the IBM CIM Object Manager daemon is installed and started by typing the following command:

```
# ps -ef --columns 1000 | grep -E "com/.ibm/.cimom/.CIMOM"
```

Here is a sample output of the above command, notice that the wrapped response line that ends with *ibmca.jar com.ibm.cimom.CIMOM*:

```
root 6322 1515 0 15:55 pts/0 00:00:00 grep -E com/.ibm/.cimom/.CIMOM
```

- b. If it is not started, you can run the **startcimom** file by issuing the following command:

```
<dest-path>/startcimom
```

Where *<dest-path>* is the destination directory where the CIM Agent is installed.

Result:

If you are able to perform all of the verification tasks successfully, then the CIM Agent has been successfully installed on your Linux system.

Related topics:

- “Configuring the CIM Agent for Linux”

Configuring the CIM Agent for Linux

This task helps you configure the CIM Agent for Linux.

Context:

This task is performed after successful installation of the ESS CIM Agent . The installation program performs the following tasks:

- Copies the directories and files from the TotalStorage CIM Agent API CD-ROM into the chosen destination directory.
- Updates some script files using the selected destination.
- Compiles the MOF files, creates the persistent sub-directory in the destination directory.
- Generates a certificate called **truststore** in the destination directory, only for the server.

Note: For a client installation, you must copy this certificate from the server and install it in the destination directory on the client machine.

Steps:

Perform the following steps to configure the ESS CIM Agent:

1. Configure the CIMOM as follows:

- Run the **setuser** file:

```
# <dest-path>/setuser
```

Where *<dest-path>* is the destination directory where the CIM Agent is installed.

Here is a sample output of the above command:

```
# ./setdevice
Application setdevice started in interactive mode
To terminate the application enter: exit
To get a help message enter: help
```

- Obtain a username and password for each user that can manage the CIMOM.

Run the following command for each user:

```
>>>adduser <username> <password>
```

Here is a sample output of the above command:

```
An ess provider entry for IP x.xx.xxx.xx successfully added
```

Use the **exit** command to exit the CIMOM configuration program.

A file named **cimom.passwd** is created in the destination directory.

2. Configure the CIM Agent API as follows:

- Run the **setdevice** file:

```
# <dest-path>/setdevice
```

Where *<dest-path>* is the destination directory where the CIM Agent is installed.

- Obtain an IP address, username, and password for each ESS Device that the CIM Agent is to manage.

Run the following command for each ESS:

```
>>>address <IP address> <username> <password>
```

Use the **exit** command to exit the CIMOM configuration program.

A file named **provider-cfg.xml** is created in the target directory.

Result:

If you are able to perform all of the configuring tasks successfully, then the CIM Agent has been successfully installed on your Linux system.

Related topics:

- “Running the CIM Agent on Linux”

Configuring the CIM Agent to run in unsecure mode on Linux

Some vendor software may not be capable of communicating with IBM's ESS CIM Agent in a secure fashion. If you wish to, you can still use this vendor software by configuring the ESS CIM Agent to run with only basic user and password security. These are the steps:

1. Enter the **stopcimom** command to stop the CIMOM.
2. Find the `cimom.properties` file in the target directory and edit it with a tool such as notepad:

```
Set ServerCommunication=HTTP
Set Port=5988
Set DigestAuthentication=False
```

3. Restart the CIMOM.

Running the CIM Agent on Linux

This section will assist you in running the CIM Agent on your Linux operating system.

Prerequisites:

Before using the CIM Agent, you can verify that the CLI for ESS 2105 software has network connectivity to the Enterprise Storage Server by issuing the following command from a Command Prompt window:

```
# /<esscli-path>/esscli -u essuser -p esspass -s 9.111.111.111 list server
```

Where *<esscli-path>* is the path where CLI for ESS 2105 is installed and where:

- *essuser* is an ESS Specialist userid
- *esspass* is the ESS Specialist password
- *s* is the IP address of the ESS machine

The following is an example of the output from the previous command:

```
Thu Oct 17 10:19:58 PDT 2002 IBM ESSCLI 2.1.0.0
```

Server	Model	Mfg	WWN	CodeEc	Cache	Nvs	Racks
2105.FCA75	F20	IBM	5005076300C00C4B	1.5.2.43	8GB	384MB	1

The installation, verification, and configuration of the CIM Agent need to be completed prior to this task. The tasks being performed are starting and verifying the Service Location Protocol (SLP) daemon, and starting and verifying the CIMOM.

Steps:

Perform the following steps to execute the TotalStorage CIM Agent:

1. Start the SLP daemon, if it is not started, run the following command from a separate Command Prompt window.

```
# /etc/init.d/slpd start
```

This session will remain active until you stop it. You should keep it running as long as the CIM Agent is running.

2. If the CIMOM is not started, start it by performing the following:

- Run the startcimom file in a Command Prompt window

```
# <dest-path>/startcimom
```

Where *<dest-path>* is the destination directory where the CIM Agent is installed.

- The default is to start the secure CIMOM. It will register itself with SLP and accept requests on Port 5989.

3. Verify the SLP registration.

Run the following command from another window:

```
# <dest-path>/checkcimreg
```

Or, as an alternative:

```
# export LD_LIBRARY_PATH=<dest-path>/slp
# <dest-path>/slp/slptool findsrvs service:wbem
```

You should see the CIMOM you started, registered as a service.

4. The entire CIM 2.7 schema reported is displayed. See Figure 21.

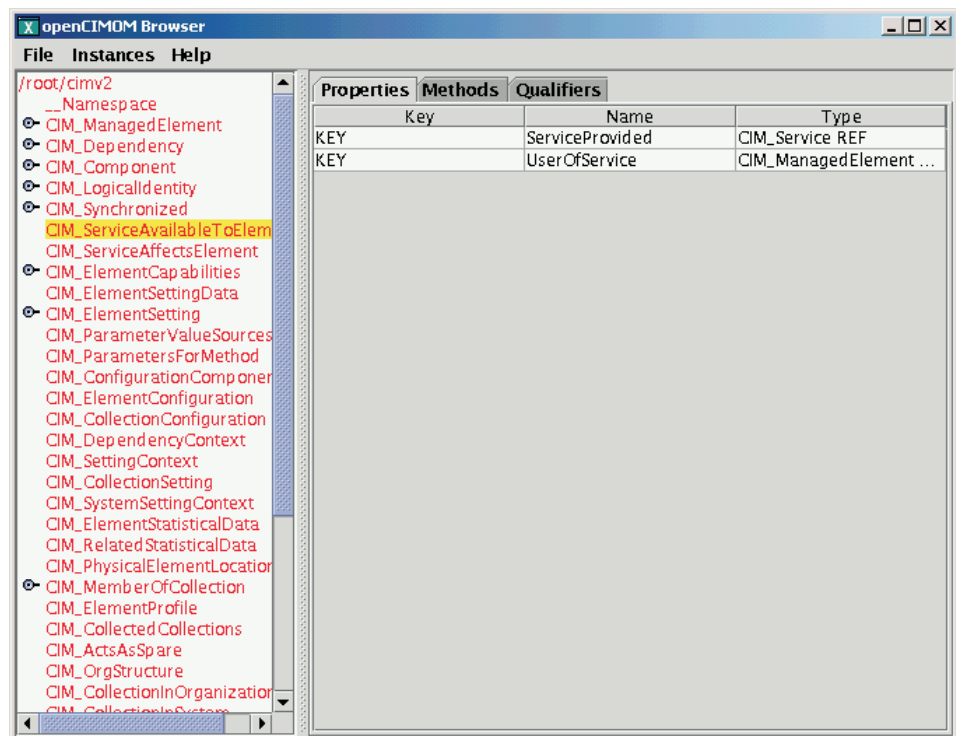


Figure 21. CIM 2.7 schema reported. CIM 2.7 schema reported

5. Enter the following command to locate all WBEM services (for example, CIMOMs) in the local network, display them and then call the CIMOMs on the local machine only requesting information about whether the CIMOMs know of any ESS.

Run the following command from a Command Prompt window:

```
# <dest-path>/verifyconfig -u <user> -p <password>
```

Where *<user>* and *<password>* are the user and password of a cimom userid created with **setuser** command.

Here is a sample output:

```
# verifyconfig -u guest -p guest
Verifying configuration of ESS CIM Agent...
Communicating with SLP to find WBEM services...
1 WBEM services found
  host=kirchhofer, port=5989
Connecting to ESS CIM Agent, host=kirchhofer, port=5989
Found 1 IBMTSESS_StorageSystem instances
Verification Successful
```

Result:

This completes the execution of the TotalStorage CIM Agent.

Removing the CIM Agent for Linux

This task takes you through the steps necessary to remove the CIM proxy API from your Linux system.

Steps:

Perform the following steps to remove the CIM proxy API:

1. Log on as a user with root authority.
2. Stop the **IBM CIM Object Manager Service** and **Service Location Protocol services** if they are started. By performing the following commands. The screens following the commands show example results from the preceding command.

```
# ps -ef --columns 1000 | grep -v "com/.ibm/.cimom/.CIMOM"
```

```
root 52858 1 0 0ct 16 pts/7 2:29 /opt/IBM/ICAT/ibmjava131/jre/bin/
java -Xms128m -Xmx256m -cp /opt/IBM/ICAT/lib/xml4j-4_0_5/xercesImpl.jar:
/opt/IBM/ICAT/lib/xml4j-4_0_5/xmlParserAPIs.jar:/opt/IBM/ICAT/ibmjava131/
jre/lib/ext/ibmjisse.jar:/opt/IBM/ICAT/lib/JCE/IBMJCEfw.jar:/opt/IBM/ICAT/
lib/JCE/IBMJCEProvider.jar:/opt/IBM/ICAT/lib/JCE/local_policy.jar:/opt/
IBM/ICAT/lib/JCE/US_export_policy.jar:/opt/IBM/ICAT/lib/JCE/ibmpkcs.jar:
/opt/IBM/ICAT/ibmjava131/jre/lib/rt.jar:/opt/IBM/ICAT/ibmica.jar com.
ibm.cimom.CIMOM
root 62884 40012 3 18:29:22 pts/7 0:00 grep cimom
```

If the CIMOM daemon is started enter the following command to stop it:

```
# <dest-path>/stopcimom
```

where *<dest-path>* is the directory where the CIMOM daemon resides.

3. Check to see if the SLP daemon is started by issuing the following command:

```
# ps -ef | grep slpd
```

```
daemon 61026 1 0 Oct 16 - 0:18 /opt/IBM/ICAT/slp/slpd
root 62884 40012 3 18:29:22 pts/7 0:00 grep slpd
```

```
# kill 61026
```

4. You can invoke the uninstall program by issuing the following commands:

```
# <dest-path>/_uninst/uninstaller -silent
```

If the CIMOM Agent server or the Service Location Protocol are still running the uninstall will fail. In this case you need to stop both of those services or run the uninstall program with `-G stopProcessesResponse="yes"`:

```
# <dest-path>/_uninst/uninstaller -silent -G stopProcessesResponse="yes"
```

The removal program messages are logged in `/tmp/cimagent/install.log`.

5. Run the uninstall program from the `_uninst` subdirectory of the `<dest-path>`, by typing the following commands:

```
# cd <dest-path>/_uninst
# uninstaller
```

Note: `<dest-path>` is the target directory where CIM Agent is installed.

6. It is possible that the, the uninstaller launcher was not created during the CIM Agent installation. If this is the case, run the following command:

```
# <dest-path>/ibmjava131/jre/bin/java -jar <dest-path>/_uninst/uninstall.jar
```

Note: `<dest-path>` is the target directory where CIM Agent is installed. It is also possible that the `uninstall.jar` file was not created. If this is the case, you can remove the CIMOM Agent rerunning the installer, `setuptools`, which will clean the previous installation.

The program detects if the Service Location Protocol (SLP) and IBM CIM Object Manager services are running or not and displays this information:

- If they are running, the program warns you that it will automatically stop them if you select **Next**. You should stop them manually and then select **Next**.

Note: You must be careful if you have other applications that use SLP service and, in this case, you must stop these applications before stopping SLP service, because during the uninstall the SLP service will be deleted. See Figure 22 on page 48.

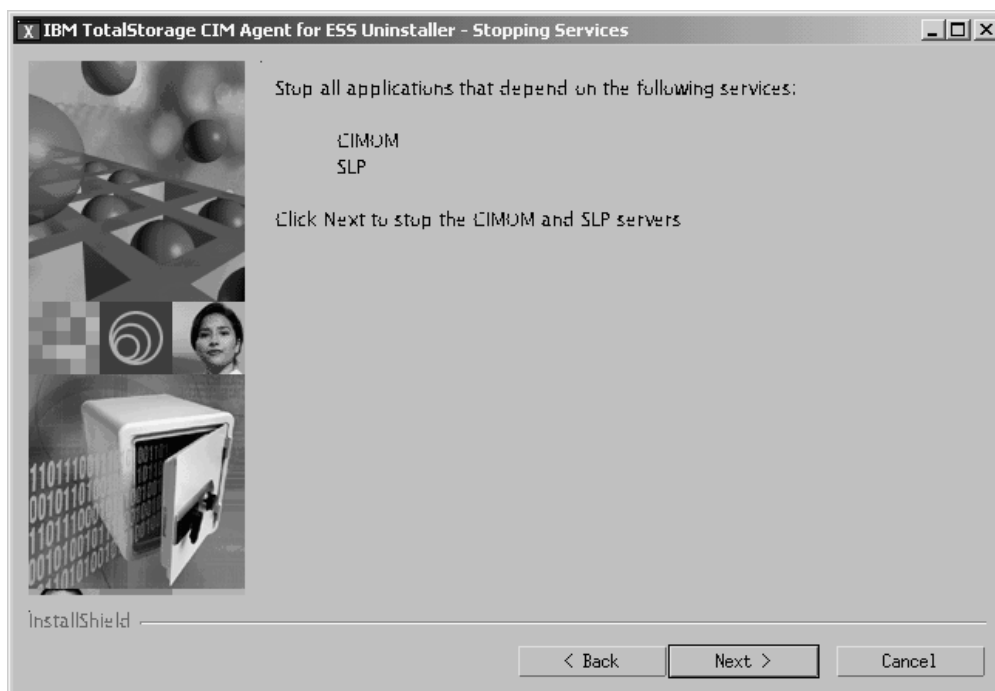


Figure 22. Stop SLP and CIMOM window

- If neither the SLP service nor the CIMOM service is running, the removal process continues with a **Preview** window.
7. The **Welcome** window opens. Click the **Next** button to continue with the removal process, or click **Cancel** to exit the uninstall program. See Figure 23

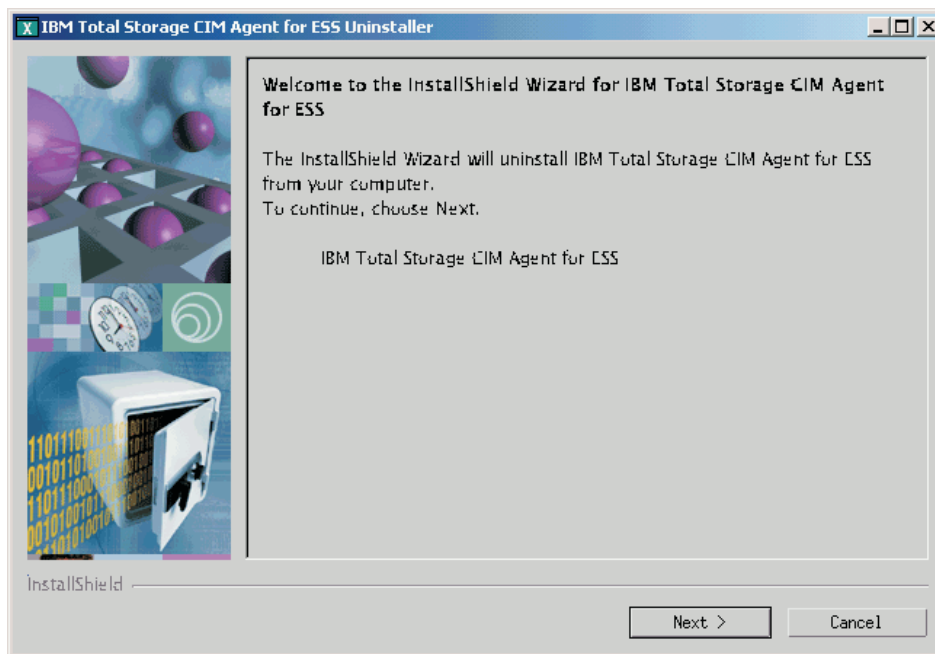


Figure 23. Welcome window

8. The **Preview** window opens, displaying the location of the product to be removed. Click the **Remove** button to continue with the removal process, or click **Cancel** to exit. See Figure 24.

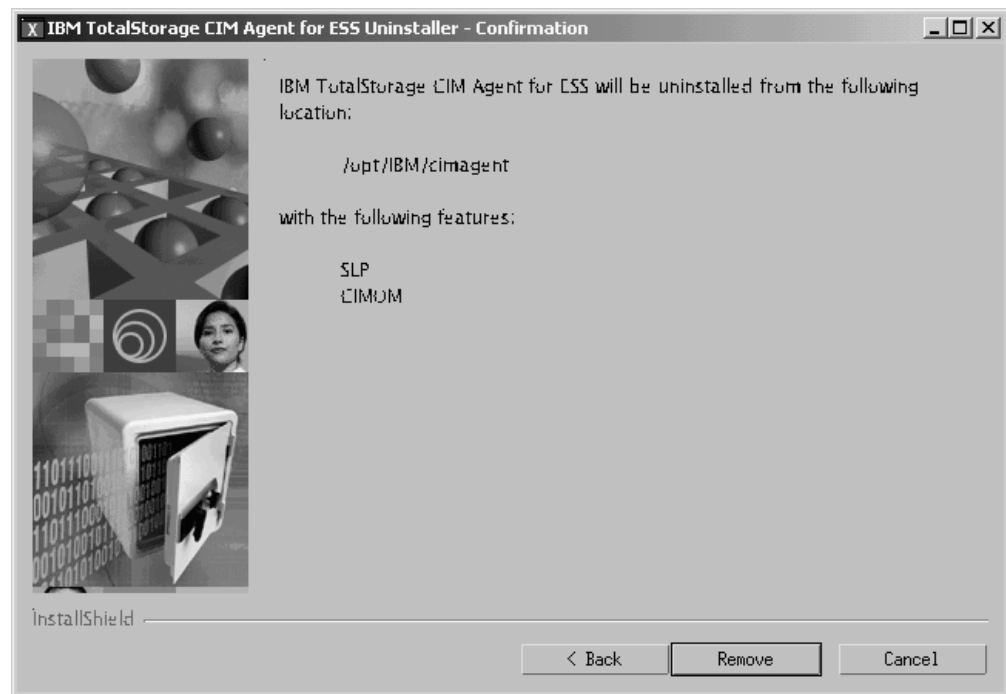


Figure 24. Preview Window

9. Wait for the program to remove CIM Agent product. The **Uninstallation Complete** window opens displaying information about the result of the removal process (successful or failed). See Figure 25 on page 50.

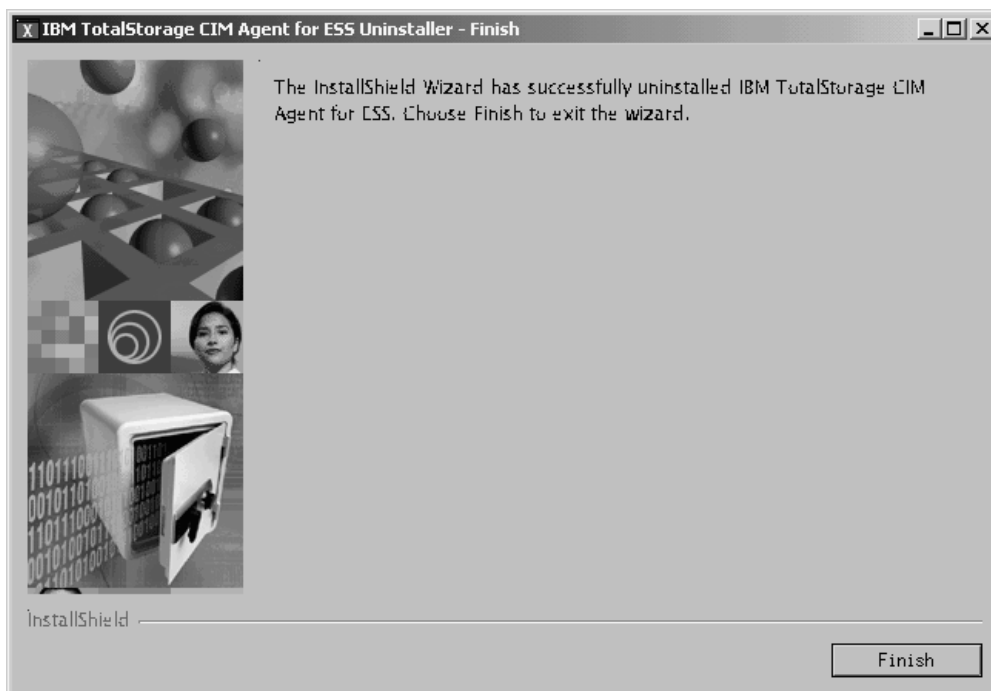


Figure 25. Summary Window

Select the **Finish** button of this window to end the removal process.

Post-processing requirements:

The CIM Agent removal process does not remove configuration files, logs, and similar files that are created during or after the installation process. They are located in the destination path where CIM Agent component was installed. For example, the default target directory is **/opt/IBM/cimagent**.

Remove the directory and all of its contents (especially if you plan to reinstall CIM Agent).

Note: If you want to keep the old configuration files, before removing them from the installation destination path, save them in another location on your system to restore them later.

To remove the directory cimagent you must type the following command:

```
# rm -Rf /opt/IBM/cimagent
```

Note: The recursive remove is used in the example because the CIM Agent has a deep directory structure. The recursive remove is very powerful and dangerous. You need to use the fully qualified directory name.

Chapter 4. CIM Agent for Windows

This chapter includes an overview of the installation process and instructions for installing and configuring the CIM Agent on a Windows 2000 operating system.

Installation overview for Windows

This section provides an overview of the installation and configuration of the CIM Agent on a Windows 2000 operating system. You should have some knowledge of how to administer Windows 2000 operating system before you install the CIM Agent. You should also become familiar with the command that you use during installation and configuration of the CIM Agent. See Chapter 5, "CIM Agent commands", on page 73 for information about the commands.

The following list of installation and configuration tasks are in the order in which they should be performed:

1. Before you install the CIM Agent for Windows, you should check the hardware and software requirements listed in "CIM Agent installation requirements" on page 4.
2. Install the prerequisite ESS CLI software, as instructed in "Installing the ESS CLI for Windows".
3. You can choose to install the CIM Agent either in graphical mode with the help of an installation wizard or in unattended mode (also known as silent mode), which involves customizing a response file and issuing a command.
 - a. If you want to install the CIM Agent in graphical mode perform the instructions in "Installing the CIM Agent on Windows in graphical mode" on page 53.
 - b. If you want to install the CIM Agent in unattended mode perform the instructions in "Installing the CIM Agent on Windows in unattended (silent) mode" on page 60.
4. Configure the CIM agent for Windows by performing the instructions in "Configuring the CIM Agent for Windows" on page 63. You might want to revisit this section in the future as you add, change, or delete CIMOM authentication and ESS information.
5. Verify connection to your ESS by performing the instructions in "Verifying connection to the ESS" on page 65.
6. Instructions for removing the CIM Agent are also provided in "Removing the CIM Agent for Windows" on page 66. You only need to perform this optional task if you get errors during installation verification or if the CIM Agent did not set the environment variables.

Installing the ESS CLI for Windows

Before you install the CIM Agent you must install the IBM TotalStorage Enterprise Storage System (ESS) Command Line Interface (ESS CLI). The CIM Agent installation program checks your system for the existence of the ESS CLI and reports that it cannot continue if the ESS CLI is not installed.

Steps:

Perform the following steps to install the ESS CLI for Windows:

1. Insert the CD for the ESS CLI in the CD—ROM drive.

The ESS CLI software can be found on the CD with the label *ESS CLI*.

2. Select the correct software from the *ESS CLI* CD, as it contains two products with similar names.

The product you need can be found in the winCL\IBMesscli subdirectory of the *ESS CLI* installation CD. The product found in the winCL\IBMcli is **not** the required prerequisite for the CIM Agent.

Note: The CIM Agent supports ESS CLI level 2.1.0.8 or later.

3. Respond **Yes** when you are prompted by the "CLI for ESS 2105" installation program to have it set the system path for you.
4. Verify that the ESS CLI is installed.

You can determine if the ESS CLI is installed by reviewing the installed programs on your Windows 2000 system. Follow the steps to review the installed programs:

- a. Go to **Start** → **Settings** → **Control Panel**.
- b. Double click the **Add/Remove Programs** icon.
- c. Verify that there is an "IBM 2105 CLI" entry.
5. Verify the ESS CLI connectivity to the ESS. For example, from a command prompt window issue the following command:

```
esscli -u essuser -p esspass -s 9.111.111.111 list server
```

The following response is displayed, where

essuser is an Enterprise Storage Server Specialist userid
 esspass is the Enterprise Storage Server Specialist password for that userid
 9.111.111.111 is the IP address of the Enterprise Storage Server

```
C:\>esscli -u username -p userpass -s 9.111.111.111 list server
Thu Oct 17 10:15:50 PDT 2002 IBM ESSCLI 2.1.0.8
```

Server	Model	Mfg	WWN	CodeEc	Cache	Nvs	Racks
2105.22496	800	075	5005076300C00C4B	1.5.2.43	8GB	384MB	1

```
C:\>
```

Post-processing requirements:

If the CLI for ESS 2105 is installed but the environment variables are not set properly, the program might display an error message, for example:

- C:\Program Files\IBM 2105 CLI>esscli
 'esscli' is not recognized as an internal or external command, operable program or batch file.
- C:\Program Files\IBM 2105 CLI>esscli
 esscli: No value is specified for the <INSTALL> system variable.
- C:\Program Files\IBM 2105 CLI>esscli
 esscli: The CLI.CFG file is not found or accessible.

If you receive error messages similar to these, use the **Add/Remove Programs** Windows facility to remove the ESS CLI, then reinstall it and accept the option to set the PATH (3).

Installing the CIM Agent on Windows in graphical mode

This task will assist you with the installation of the CIM Agent in your Windows environment using the graphical mode. If you choose to install the CIM Agent in unattended mode (also known as silent mode) skip this section and follow the instructions in "Installing the CIM Agent on Windows in unattended (silent) mode" on page 60. You must satisfy all prerequisites listed in "CIM Agent installation requirements" on page 4 before starting the installation.

Steps:

You should have some knowledge of how to administer a Windows 2000 operating system before beginning the installation of the CIM Agent.

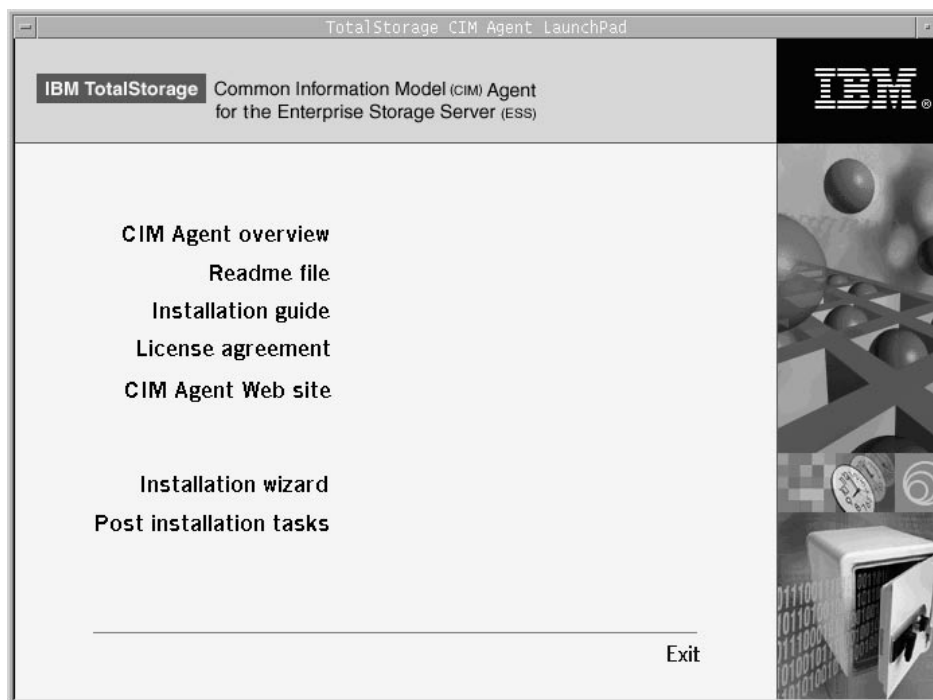
Perform the following steps to install the CIM Agent on your Windows system:

1. Log on to your system as the local administrator. For example the "Administrator" user.
2. Insert the *Common Information Model Agent* CD into the CD-ROM drive.
The CIM Agent program should appear within 15 - 30 seconds if you have autorun mode set on your system. If the LaunchPad window does not appear, perform the following steps:
 - a. Use a Command Prompt or Windows Explorer to change to the W2K directory on the CD.
 - b. If you are using a Command Prompt window, type:
LaunchPad
 - c. If you are using Windows Explorer, double-click on the **LaunchPad.bat** file.

Note: If you are viewing the folder with the Windows Explorer with the option selected to hide file extensions for known file types, find the LaunchPad file within the file type of "MS-DOS Batch File".

3. The following options are displayed when the LaunchPad window opens (see Figure 26 on page 54.):
 - Product Overview - offers information about the CIM Agent
 - ReadMe File - offers any last minute product information that did not make it into this installation guide
 - Installation Guide - offers instructions on how to install the CIM Agent (a softcopy of this document)
 - Product Web Information - offers information from the product web site
 - License Information - offers information about the license of the CIM Agent
 - Install the Product - starts the CIM Agent installation program
 - Exit - exits the CIM Agent installation

Figure 26. LaunchPad window



4. Click the **ReadMe File** from the LaunchPad window or from the **README.txt** file located in the doc or W2k directory on the Common Information Model Agent CD to check for information that might supersede the information in this guide.
5. Click **Install the Product** from the LaunchPad window to start the installation.

Note: The LaunchPad is independent of the installation wizard and remains open (behind the installer) during the installation. This allows you access to product information after the installation has started. Click **Exit** to close the LaunchPad.

6. There may be a slight delay while the software loads on your system until a DOS prompt window opens to display the following message:

```

Initializing InstallShield Wizard...
Preparing Java (tm) Virtual Machine .....
.....

```

7. The Welcome window appears suggesting what documentation you should review prior to installation. (See Figure 27.) Click **Next** to continue, or click **Cancel** to exit the installation.

Figure 27. Welcome window



8. The License Agreement window appears. (See Figure 28 on page 56.) Read the license agreement information. Select **I accept the terms of the license agreement**, then click **Next** to accept the license agreement. Otherwise, keep the selection **I do not accept the terms of the license agreement** (it is the default) and click **Cancel** to exit the installation.

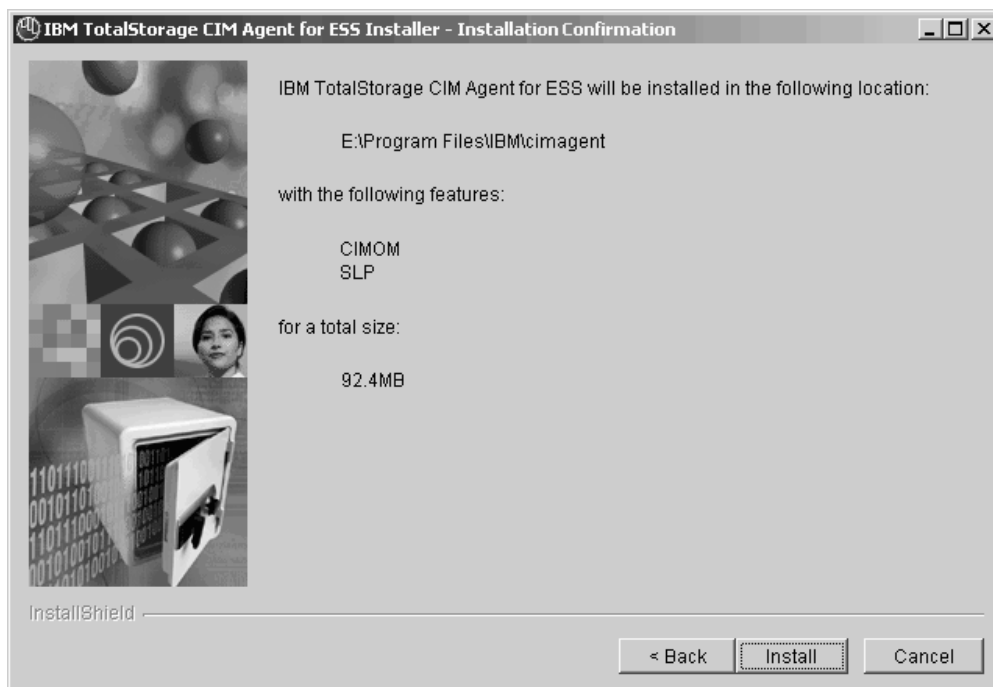


Figure 28. License Agreement window

9. The installation program performs a verification test to determine whether or not your machine has met the installation requirements:
 - If you have a Service Location Protocol (SLP) installed on your operating system that is different from the SLP installed by the CIM Agent installer, the program displays an error and asks you to stop the installation and remove this SLP service from the system.
 - If the ESS CLI Client is not installed or is installed with a down-level version, an error is displayed and the installation program exits. If the ESS CLI is installed and its version is 2.1.0.8 or later, the installation program updates the PATH environment variable to include the INSTALL environment variable value.
 - The installation program checks if a version of the CIM Agent is already installed. If the CIM Agent is installed, it checks if the Service Location Protocol service and the IBM CIM Object Manager service are started.

Continue the installation program by clicking **Next**, or click **Cancel** to exit the installation program. If you choose to exit the installation program, you must manually stop the services and applications that use these services, and restart the CIM Agent installation program. (You can save the old configuration by selecting the check box that you see on the screen.)

10. The Select Destination window opens. (See Figure 29 on page 57.) Click **Next** to accept the default directory, click **Browse** to select a different directory for installation. Click **Next** to continue after you select a directory or click **Cancel** to exit the installation program.

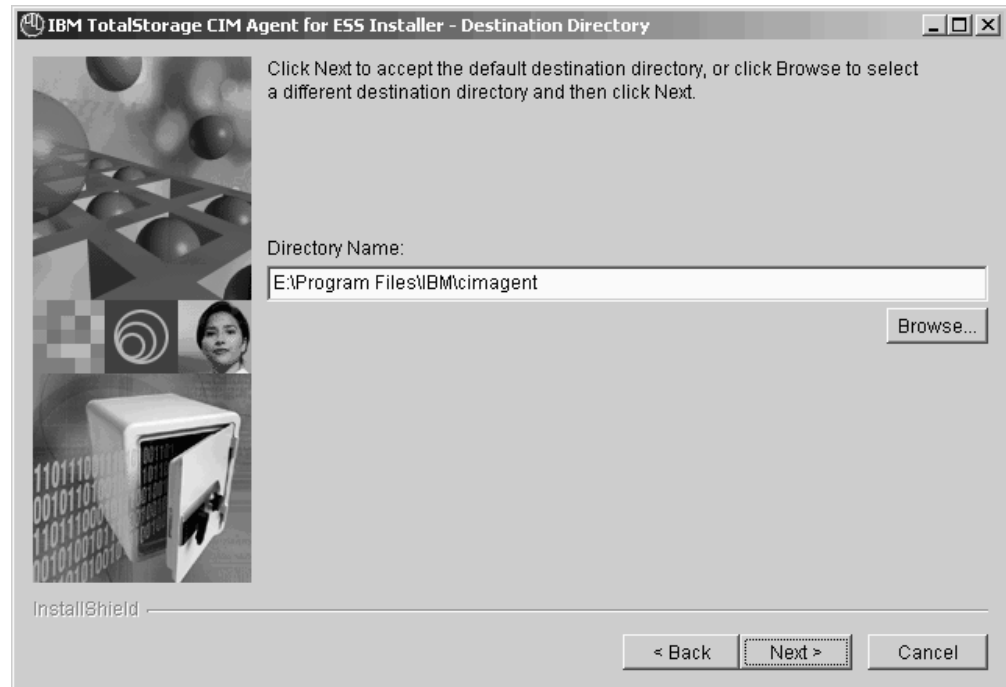


Figure 29. Select Destination window

Note: If the program detects insufficient space for the CIM Agent installation in the chosen destination, an error message appears. You must stop the installation program by clicking **Cancel**, free some space in that destination, and restart the installation program. You can also go back by clicking **Back**, and choose another destination for the product.

11. The Preview window opens. (See Figure 30 on page 58.) Click **Install** to confirm the installation location and file size. Click **Cancel** to exit the installation program.

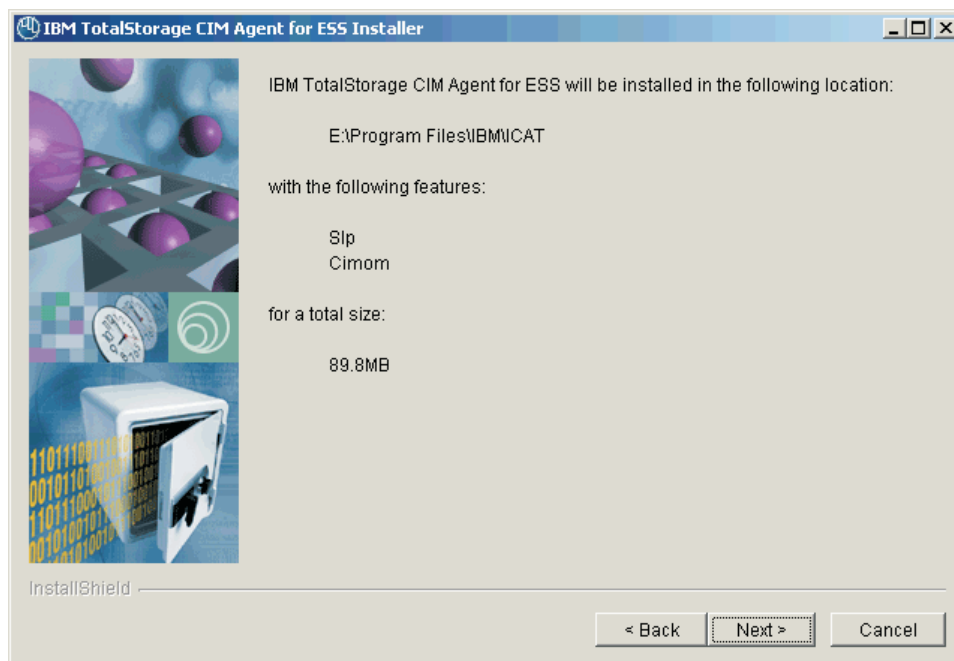


Figure 30. Preview window

12. The Installation Progress window opens indicating how much of the installation has been completed. (See Figure 31 on page 59.) Installation usually takes 3 - 10 minutes depending on the configuration of your machine. You can click **Cancel** to exit the installation program.

Note: If you click **cancel** a popup window appears asking you to confirm the cancellation of the installation program: "Cancel the current operation? **Yes No**". Be aware that if you confirm the cancellation by clicking Yes, the information you entered or selected in previous windows is not saved. You must start the installation again from the first step.

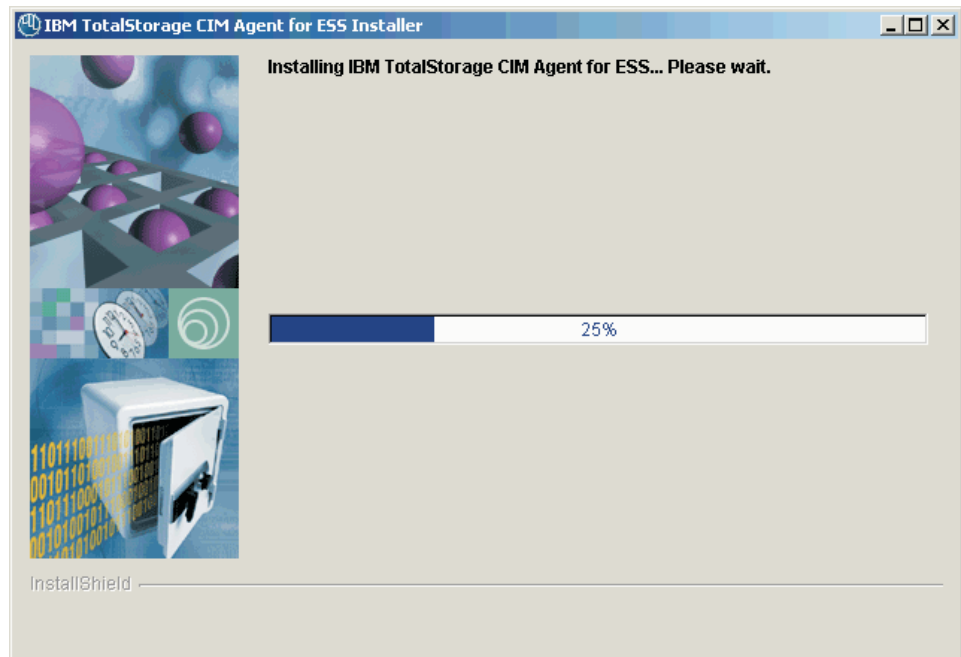


Figure 31. Installation Progress window

13. The Installation Complete window opens. (See Figure 32.) Before proceeding, review the log file for any possible error messages. The log file is located in xxx\logs\install.log, where xxx is the destination directory where the CIM Agent for Windows was installed. The install.log contains a trace of the actions performed by the installer. Click **Finish** to exit the Installshield wizard.

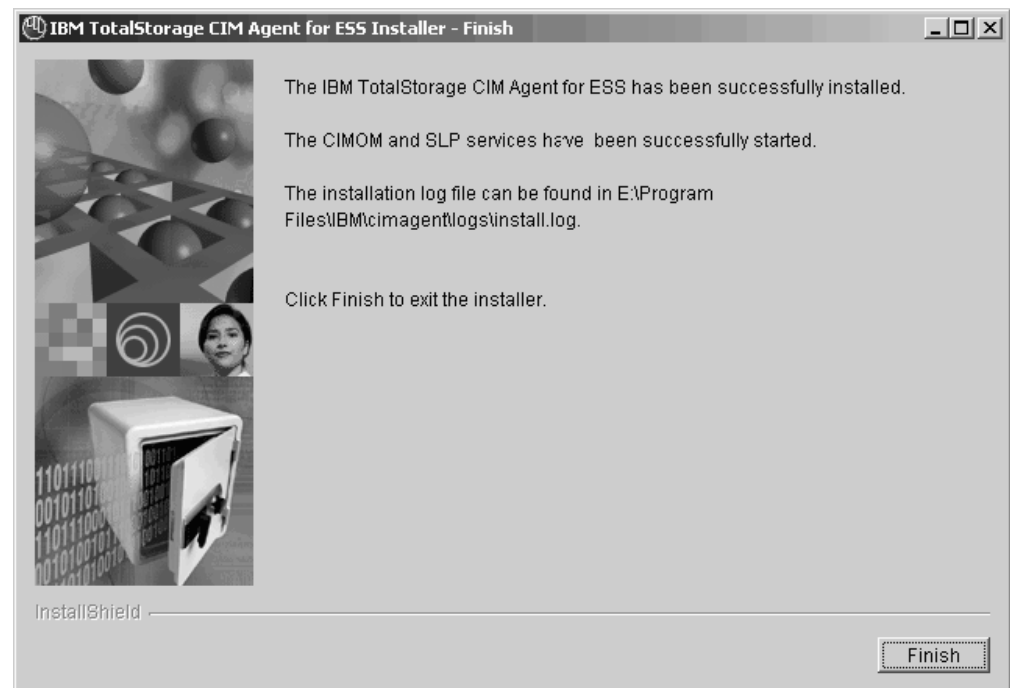


Figure 32. Installation Complete window

Note: Ordinarily, a reboot is not necessary during or after the installation of CIM Agent. However, the installation program may determine during its run that reboot is necessary. If the installation wizard advises you that a reboot is necessary, then reboot your system. After you reboot the system, the installation program continues with the installation.

14. Exit the LaunchPad program by clicking **Exit** on the LaunchPad window.
15. Verify the installation by performing the tasks in “Verifying the CIM Agent Windows installation” on page 63.

Related topics:

- “Installing the CIM Agent on Windows in unattended (silent) mode”
- “Verifying the CIM Agent Windows installation” on page 63
- “Configuring the CIM Agent for Windows” on page 63
- “Removing the CIM Agent for Windows” on page 66

Installing the CIM Agent on Windows in unattended (silent) mode

This task will assist you with the installation of the CIM Agent in your Windows environment using the unattended (silent) mode. This method of installation involves customizing a response file and issuing a command from a Command Prompt. You must satisfy all prerequisite listed in “CIM Agent installation requirements” on page 4 before starting the installation.

Context:

The silent install option allows you to run installation unattended. You can create a standard response file to ensure the product is installed consistently on multiple systems. The **responsefile** file is a template located on the CIM Agent CD-ROM that you must modify.

Steps:

Perform the following steps to install the CIM Agent in your Windows environment using the unattended mode:

1. Log on as local administrator user.
2. Locate the **responsefile** file on your CIM Agent CD-ROM in the W2K directory.
3. Using Windows Explorer or a command prompt, copy the **responsefile** file to your hard drive.
4. Customize the **responsefile** file:
 - a. Follow the directions in the **responsefile** template.
 - b. Modify the default parameters in the **responsefile** file with your desired values in a text editor:
 - Remove the # character from the beginning of the command line and edit the lines to use values other than the defaults. You **must** enclose all values in double quotation marks (").
 - The variable `<product.installLocation>` is used to define the default location where the product will be installed. To use another destination location, uncomment the corresponding line and replace this default location with the desired location. For example: `c:\Program Files\IBM\cimagent`.
 - The variable `<stopProcessesResponse>` tells the program whether or not to stop the SLP and the IBM CIM Object Manager processes if they are

active. By default this variable value is "no" and if you do not change this value, the program will abort the installation if these processes are active. To enable the automatic stop of the SLP and CIMOM, uncomment the corresponding line and change it's value to "yes".

- The variable `<checkPrerequisite>` checks the prerequisites. To disable checking the prerequisites uncomment the corresponding line and change it's value to "no". The variable checks to see if the ESS CLI is installed on the system.
 - The variable `<startUpgrade>` enables the installation of the product with the same or a higher version. To do this uncomment the corresponding line and change it's value to "yes".
 - The variable `<saveConfiguration>` saves the configuration files on a drive for automatic recovery at a later time. If you do not want to save the configuration files, uncomment the corresponding line and change it's value to "no".
5. Save the modifications to the **responsefile** file. Save the customized responsefile in some drive other than the CD drive to avoid confusing it as a duplicate of the original file. Make sure the file is saved without a file extension such as .txt.
 6. Using a command prompt window, enter the following command:

```
CD <drive path> W2K\install -options <response file path>\responsefile
```

where `<drive path>` is the path of your CD-ROM drive. `<response file path>` is the path of the responsefile file that you copied in step 3 on page 60 and customized in step 4 on page 60.

7. During the installation you will see dotted lines scrolling across the screen. When the installation program ends you see the cursor.
8. Check for installation errors in the **install.log** file. After all the prerequisites checks have been performed, the log file is copied. This file can be found in the `<dest-path>\logs\` directory folder. This file is initially created in the system temporary file under the subdirectory **cimagent**. The following is an example of an **install.log** file:

```

(Nov 8, 2002 3:21:01 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing External Files (SLP Files)
(Nov 8, 2002 3:21:02 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing ICA Install SLP (Install slpd Service)
(Nov 8, 2002 3:21:03 PM), Service Location Protocol successfully installed .
(Nov 8, 2002 3:21:03 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing External Files (IBM JAVA 1.3.1 Files)
(Nov 8, 2002 3:21:51 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing External Files (CIMOM Files)
(Nov 8, 2002 3:23:47 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Bat Replace (Bat Replace)
(Nov 8, 2002 3:23:47 PM), The file replacing starts ...
(Nov 8, 2002 3:23:47 PM), The file setupCmdLine.bat successfully replaced
(Nov 8, 2002 3:23:47 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Compile Mof (Compile Mof)
(Nov 8, 2002 3:23:47 PM), CIMAgent, com.ibm.itcins.ical.CompileMof, msg1, Compile MOF files started ...
(Nov 8, 2002 3:24:36 PM), CIMAgent, com.ibm.itcins.ical.CompileMof, msg1, MOF files successfully
compiled.
(Nov 8, 2002 3:24:36 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Cert Man (Cert Man)
(Nov 8, 2002 3:24:36 PM), CIMAgent, com.ibm.itcins.ical.CertMan, msg1, Generate a certificate store
started ...
(Nov 8, 2002 3:24:51 PM), CIMAgent, com.ibm.itcins.ical.CertMan, msg1, Certificate store called truststore
successfully generated.
(Nov 8, 2002 3:24:51 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing ICA Install CIM (Install CIMOM Service)
(Nov 8, 2002 3:24:52 PM), CIMOM service successfully installed .
(Nov 8, 2002 3:24:52 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (upd_conf folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (startcimom folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (stopcimom folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (cimbrowser folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (cimomupd_conf folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (slptool folder item)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (cimomupd_conf icon)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing Desktop Icon (upd_conf icon)
(Nov 8, 2002 3:24:53 PM), CIMAgent, com.installshield.wizard.platform.win32.Win32ProductServiceImpl,
msg1, installing W32 ICA Registry Update (Registry Update)
(Nov 8, 2002 3:25:23 PM), Command to be executed : net start cimomsvr
(Nov 8, 2002 3:25:23 PM), Servers successfully started.

(Nov 8, 2002 3:25:23 PM), The install log file can be found in C:\cimagent8\logs\install.log.
You can now configure the product:
- Configure users
- Configure device communications
Return to the Launchpad application if you wish to read product information/documentation.

```

9. Close the command prompt window by typing **exit**, for example

Related topics:

- “Verifying the CIM Agent Windows installation” on page 63
- “Installing the CIM Agent on Windows in graphical mode” on page 53
- “Configuring the CIM Agent for Windows” on page 63
- “Removing the CIM Agent for Windows” on page 66

Verifying the CIM Agent Windows installation

This task describes the steps necessary to verify that your CIM Agent is installed correctly on your Windows system.

Steps:

Perform the following steps to verify your CIM Agent installation:

1. Verify the installation of the Service Location Protocol (SLP).
 - a. Verify that the Service Location Protocol is started. Select **Start->Settings->Control Panel**. Double click the **Administrative Tools** icon. Double Click the **Services** icon.
 - b. Find the **Service Location Protocol** in the Services window list. For this component, the Status column should be marked **Started** and the Startup Type column should be marked **Manual**.
 - c. If SLP is not started, right click on the SLP and select **Start** from the pop-up menu and wait for the Status column to be marked **Started**.
 - d. Do not close the Services window because you will use it to verify the IBM CIM Object Manager service.
2. Verify the installation of the CIM Agent.
 - a. Verify that the CIMOM service is started. If you closed the Services window, select **Start->Settings->Control Panel**. Double click the **Administrative Tools** icon. Double Click the **Services** icon.
 - b. Find the **IBM CIM Object Manager** in the Services window list. For this component, the Status column should be marked **Started** and the Startup Type column should be marked **Automatic**.
 - c. If the IBM CIM Object Manager is not started, right click on the **IBM CIM Object Manager** and select **Start** from the pop-up menu and wait for the Status column to be marked **Started**.
 - d. Close the Services window.
 - e. Close the Administrative Tools window.

Result:

If you are able to perform all of the verification tasks successfully, then the CIM Agent has been successfully installed on your Windows system.

Related topics:

- "Configuring the CIM Agent for Windows" for configuration information.

Configuring the CIM Agent for Windows

This task configures the TotalStorage CIM Agent for Windows after successful installation of the CIM Agent

Steps:

Perform the following steps to configure the CIM Agent:

1. Configure the CIMOM for each user who will have authority to use the CIMOM. Select **Start->Programs->IBM TotalStorage CIM Agent for ESS->Configure CIMOM Users**.

This opens a Command Prompt window running the **setuser** file. Obtain a username and password for each user that can manage the CIMOM:

- a. Run the following command for each user:

```
>>>adduser <username> <password>
```

- b. If an addition or change was successful, the program displays the verification information, for example:

```
>>>adduser mary sanjose
An entry for user mary successfully added
>>>
```

2. Repeat step 1 on page 63 for each additional username that you want to configure.
3. Close the Command Prompt window by entering a command, for example, **exit**.
4. Configure the CIM Agent for each Enterprise Storage Server device to which the CIM Agent can have access. Select **Start->Programs->IBM TotalStorage CIM Agent for ESS->Enable ESS Communications** to start the CIM Agent device configuration tool.

This opens a Command Prompt window running the **setdevice** file. Obtain an IP address, username, and password for each device, such as an ESS, that the CIM Agent is to manage:

- a. Run the following command for each device:

```
>>>address <IP address> <username> <password>
```

- b. If an addition or change was successful, the program displays the verification information, for example:

```
>>>address 117.111.222 <username> <password>
An ess provider entry for IP 117.111.222 successfully added
>>
```

5. Repeat step 4 for each additional device (an ESS) that you want to configure.
6. Close the Command Prompt window by entering a command, for example, **exit**.

Result:

If you were able to perform all of the configuring tasks successfully, then the CIM Agent has been successfully installed on your Windows system.

Related topics:

- “Configuring the CIM Agent to run in unsecure mode”
- “Verifying connection to the ESS” on page 65

Configuring the CIM Agent to run in unsecure mode

Some vendor software may not be capable of communicating with IBM's ESS CIM Agent in a secure fashion. If you wish to, you can still use this vendor software by configuring the ESS CIM Agent to run with only basic user and password security. These are the steps:

1. Stop the CIMOM from the **Start->Programs->IBM TotalStorage CIM Agent for ESS->Stop CIMOM Service**.
2. Find the `cimom.properties` file and edit it with a tool such as notepad, setting the properties as shown in the following example:

```
Set ServerCommunication=HTTP
Set Port=5988
Set DigestAuthentication=False
```


When the CIMOM starts it will accept requests over HTTP using basic authentication.

Note: To completely disable security checking, enter “Set Authorization=False ” in the cimom.properties file.

3. Restart the CIMOM from the **Start->Programs->IBM TotalStorage CIM Agent for ESS->Start CIMOM Service**.
4. Close this window by pressing any key when you are prompted by the following display:

```
The IBM CIM Object Manager service is starting .....
The IBM CIM Object Manager service was started successfully
Press any key to continue ...
```

Verifying connection to the ESS

During this task the CIM Agent software will connect to the Enterprise Storage Server that you identified in the configuration task. The connection to the Enterprise Storage Server (ESS) is through the ESS CLI software. If the network connectivity fails or the userid and password which you set in the configuration task is incorrect, the CIM Agent will be unable to connect to the ESS successfully. The installation, verification, and configuration of the CIM Agent must be completed prior to verifying the connection to the ESS.

Steps:

In order to connection to an ESS, the Service Location Protocol (SLP) service and the CIMOM service must be started. The following steps tell you how to start the SLP and CIMOM services, if they have not been started by the installation program, and run the **verifyconfig** command.

1. Verify if the SLP is active by going to **Start->Settings->Control Panel**. Double click the **Administrative Tools** icon. Double Click the **Services** icon.
 - Find the Service Location Protocol (SLP).
For this component, the Status column should be marked **Started** and the Startup Type column should be marked **Manual**.
 - If SLP is not started, right click on the **SLP** and click **Start** from the popup menu. Wait for the Status to change to **Started**.
2. Verify that the CIMOM Service is started. If the CIMOM Service is not started, go to **Start->Programs->TotalStorage CIM Agent for ESS->Start cimom service**.

This opens a Command Prompt window running the **startcimom** file. The default is to start the secure CIMOM. It will register itself with SLP and accept requests on Port 5989.

3. Close this window by pressing any key when you are prompted by the following display:

```
The IBM CIM Object Manager service is starting .....
The IBM CIM Object Manager service was started successfully
Press any key to continue ...
```

4. You can verify SLP registration by selecting **Start->Programs->TotalStorage CIM Agent for ESS->Check CIMOM Registration**. The window closes when you press any key, as instructed in the output:

```
service: wbem:http://tpc035/ 5988, 65535
press any key to continue...
```

5. Use the **verifyconfig** command to locate all WBEM services in the local network. This command verifies that you have configured and can connect to at least one device, such as an ESS.

Run the **verifyconfig** command from another Command Prompt window in the directory where the CIM Agent was installed, for example:

```
c:\program files\ibm\cimagent
verifyconfig -u user -p password
```

Where -u and -p are the userid and password that you configured to manage the CIMOM during the configuration task (step 1 on page 63).

Here is a sample output:

```
C:\program files\ibm\cimagent>verifyconfig -u guest -p guest
Verifying configuration of ESS CIM Agent...
Communicating with SLP to find WBEM services...
1 WBEM services found
  host=kirchhofer, port=5989
Connecting to ESS CIM Agent, host=kirchhofer, port=5989
Found 1 IBMTSESS_StorageSystem instances
Verification Successful
```

Result:

This completes the initial configuration of the TotalStorage CIM Agent.

Removing the CIM Agent for Windows

This optional task takes you through the steps necessary to remove the CIM Agent from your Windows system.

Steps:

Perform the following steps to remove the CIM Agent:

1. Log on to the system where CIM Agent is installed with a user name that is a local system administrator.
2. Stop the **Service Location Protocol** and **IBM CIM Object Manager** services if they are started.
 - a. Select **Start->Settings->Control Panel**. In the Control Panel window, double-click on the **Administrative Tools** icon and then double-click the **Services** icon. The Services window opens.
 - b. Stop the IBM CIM Object Manager service:
 - 1) In the Services window, scroll to **IBM CIM Object Manager**. Click on the service to select it.
 - 2) If the Status column shows Started, right-click the service, then click **Stop** on the pop-up menu.
 - c. Stop Service Location Protocol service:

Note: You must be careful if you have other applications that use SLP service. In this case, you must stop these applications before stopping SLP service, because during the removal process the SLP

service will be deleted. You must also stop the configuration utilities for the CIM Agent, if they are running.

- 1) In the Services window, scroll to **Service Location Protocol**. Click on this service to select it.
 - 2) If it is running (the Status column shows Started), right-click the **Service**, then click **Stop** on the pop-up menu.
(If you did not stop the IBM CIM Object Manager service, the system now asks if you want to stop the IBM CIM Object Manager service. Because CIM Object Manager service is dependent on the Service Location Protocol service which you just stopped, you must click **YES** to stop the CIM Object Manager service.)
 - 3) Wait for the services to stop.
 - 4) Close the Services window.
 - 5) Close the Administrative Tools window.
3. Use the Windows Add/Remove Programs facility to remove the IBM TotalStorage CIM Agent for ESS and the Service Location Protocol components.
 - a. From the Windows menu bar, click **Start -> Settings -> Control Panel**. Double click on the **Add/Remove Programs** icon.
 - b. Select the **IBM TotalStorage CIM Agent for ESS** from the list of currently installed programs and click **Remove** to remove the product. (See Figure 33.)

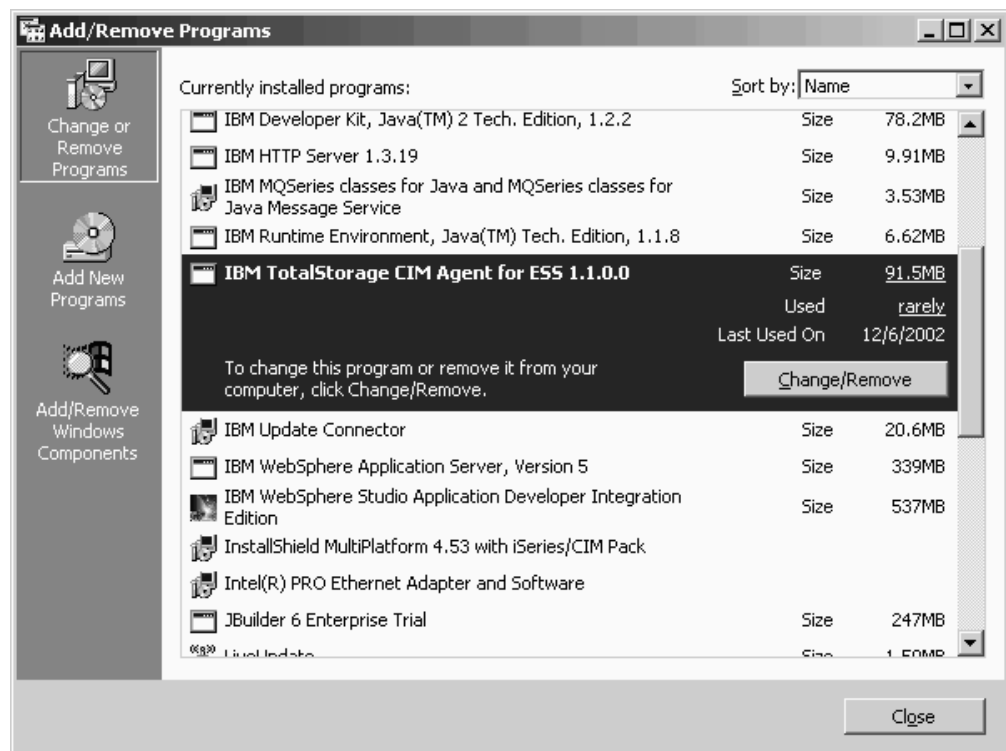


Figure 33. Add/Remove Programs Window

4. The uninstall Welcome window appears. Click **Next** to continue. (See Figure 34 on page 68.)



Figure 34. Welcome window

5. The program detects whether the Service Location Protocol (SLP) and IBM CIM Object Manager services are running and displays the following information:
 - If the services are running, the program will ask you if you want to continue with the removal process. You should consider at this point whether applications other than the CIM Agent are dependent on the services.
 - a. Click **Next** to have the program stop the services for you.
 - b. Click **Cancel** to exit the removal process if you wish to manually stop the services and any dependent applications. Instructions for stopping the services are described in step 2 on page 66. You must then restart the removal process from the Windows Add/Remove facility.
 - If the SLP service is not started, the program continues with a Preview window. (See Figure 35 on page 69.) The Preview window displays the uninstall location.
6. Click **Remove** to continue.

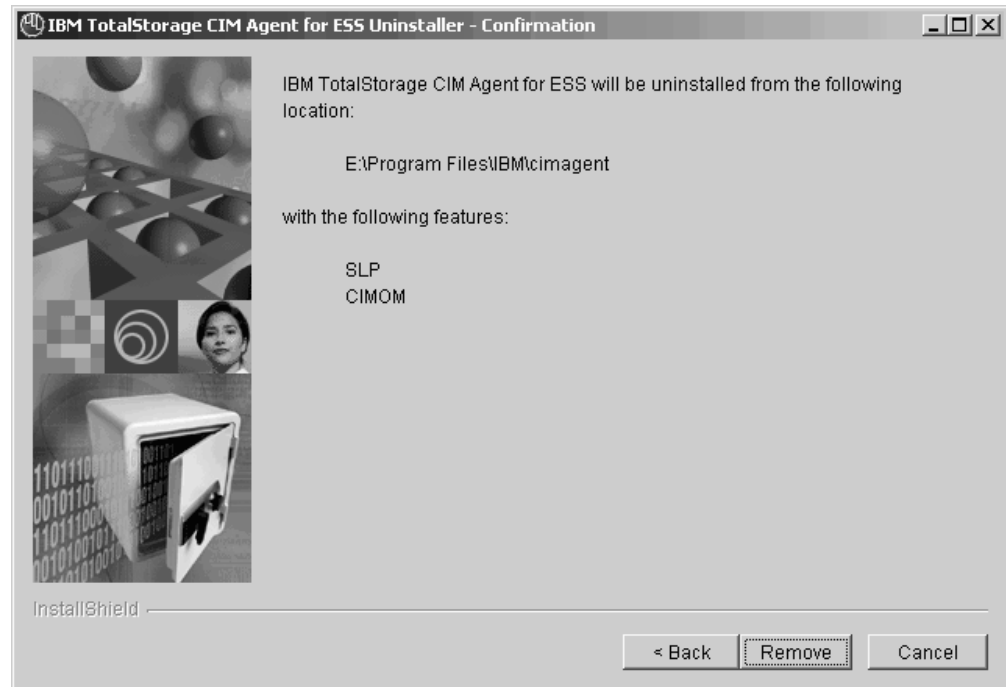


Figure 35. Preview window

7. The Uninstall Progress window opens. (See Figure 36.) Wait for the program to remove the CIM Agent product.

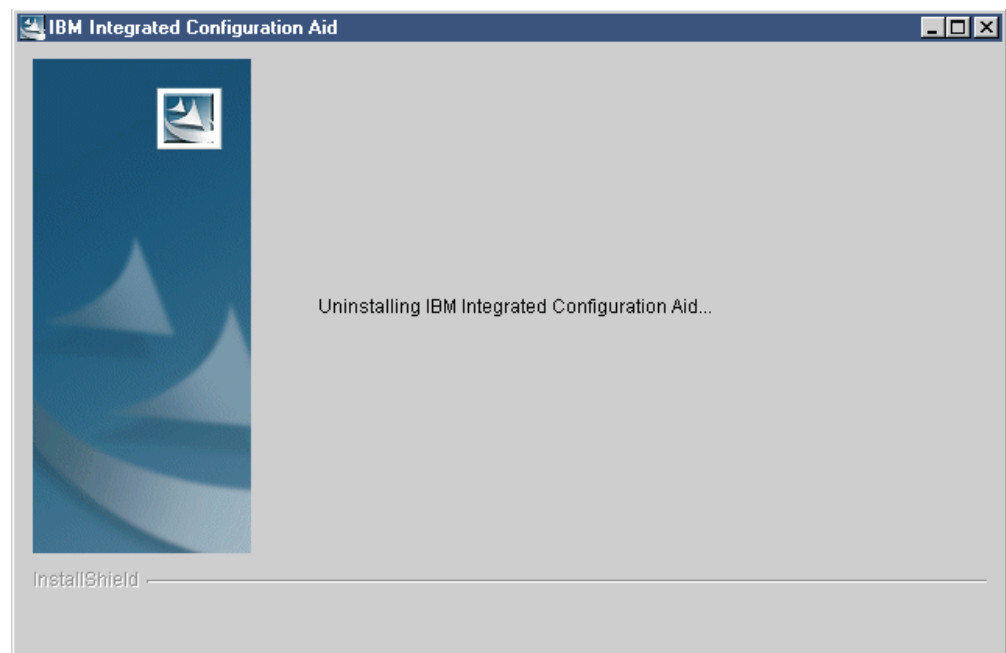


Figure 36. Uninstall progress window

8. The Uninstallation Complete window appears. (See Figure 37 on page 70.) This window opens displaying information about the result of the removal

process (successful or failed). Click **Finish** to complete the removal process and exit the wizard.

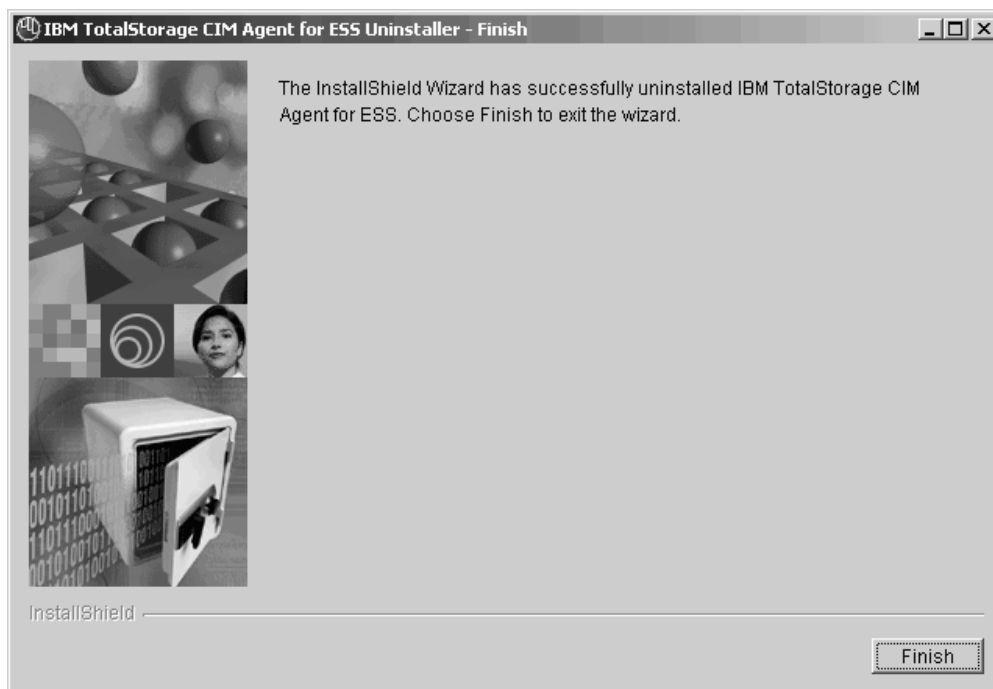


Figure 37. Uninstall Complete window

If the program could not remove some information from the system, the Restart window opens. You will need to restart your system to find and manually remove the information.

9. Close the Add/Remove Programs window.
10. Restart the system (now or later) to complete the removal process.

Post-processing requirements:

Perform the following steps to complete the removal process:

1. If the system has not been restarted since CIM Agent was removed, do so now.
2. Log on as a local administrator.
3. Remove other files and folders for CIM Agent, as the removal process does not delete configuration files, logs, and similar files that were created during or after the installation process. The files are located in the destination path, where you installed the CIM Agent. For example, the default destination path is **C:\Program Files\IBM\cimagent**. Remove the **cimagent** folder and all of its contents (especially if you plan to reinstall CIM Agent).

Note: If you want to keep the old configuration files, before removing them from the installation destination path, save them in another location on your system to restore them later.

4. Perform other cleanup tasks:
 - a. Close both the Services and the Add/Remove Program windows if you have not already done so.
 - b. Empty your Windows Recycle Bin to reclaim the disk space that was free during the removal process.

Related topics:

Chapter 5. CIM Agent commands

This chapter provides a reference to the commands that you use during installation and configuration of the CIM Agent on the UNIX, Linux, and Windows platforms. This chapter is divided into the following sections:

- Configuration commands
 - setuser
 - setdevice
- Operational commands
 - startcimom (UNIX only)
 - stopcimom (UNIX only)
- Utility commands
 - mkcertificate
 - slpd (UNIX only)
 - verifyconfig

Before you use the commands, see Chapter 4, “CIM Agent for Windows”, on page 51 for information about how to install, configure, and enable the CIM Agent on a Windows 2000 host system.

Before you use the commands, see Chapter 2, “CIM Agent for AIX”, on page 7 for information about how to install, configure, and enable the CIM Agent on an AIX host system.

Before you use the commands, see Chapter 3, “CIM Agent for Linux”, on page 29 for information about how to install, configure, and enable the CIM Agent on a Linux host system.

Conventions used in this chapter

This section describes the notational conventions that are used in this chapter for the syntax diagrams.

Syntax diagrams

A syntax diagram uses symbols to represent the elements of a command and to specify the rules for using these elements. This section shows you how to read the syntax diagrams that represent the various commands. In doing so, it defines the symbols that represent the command elements.

Main path line



Begins on the left with double arrowheads (>>) and ends on the right with two arrowheads facing each other (><). If a diagram is longer than one line, each line to be continued ends with a single arrowhead (>) and the next line begins with a single arrowhead. Read the diagrams from left-to-right, top-to-bottom, following the main path line.

Keyword

►► *esscli* ◄◄

Represents the name of a command, flag, parameter, or argument. A keyword is not in italics. Spell a keyword exactly as it is shown in the syntax diagram.

Required keywords

►► *-a* *—AccessFile* ◄◄
 -u *—UserName* *—p* *—Password* ◄◄

Indicate the parameters or arguments you must specify for the command. Required keywords appear on the main path line. Mutually exclusive required keywords are stacked vertically.

Optional keywords

►► *-h*
 -help
 -? ◄◄

Indicate the parameters or arguments you can choose to specify for the command. Optional keywords appear below the main path line. Mutually exclusive optional keywords are stacked vertically.

Default value

►► *protocol* *==* *FCP*
 FICON ◄◄

Appears above the main path line.

Repeatable keyword or value

►► *newports* *==* *ALL*
 PortId1,PortId2,... ◄◄

Represents a parameter or argument that you can specify more than once. A repeatable keyword or value is represented by an arrow returning to the left above the keyword or value.

Variable

►► *AccessFile* ◄◄

Represents the value you need to supply for a parameter or argument, such as a file name, user name, or password. Variables are in italics.

Space separator

►► *-u* *—Userid* *—p* *—Password* ◄◄

Adds a blank space on the main path line to separate keywords, parameters, arguments, or variables from each other.

Quotation mark delimiters

►—d— —"—ess—=*EssId*— —host—=*'Host Name'*— —profile—=*ProfileName*—"————►

Indicates the start and end of a parameter or argument that contains multiple values. Enclose one or more name-value pairs in a set of double quotation marks for a particular parameter or argument. If the value of a parameter or name-value pair contains a blank or white space, enclose the entire value in a set of single quotation marks.

Equal-sign operator

►—"—ess—=*EssId*— —profile—=*ProfileName*—"————►

Separates a name from its value in a name-value pair.

Syntax fragment

►—| Fragment name |————►

Fragment name:

|—(*fragment details*)—|

Breaks up syntax diagrams that are too long, too complex, or repetitious. The fragment name is inserted in the main diagram, and the actual fragment is shown below the main diagram.

Special characters

The following special characters are used in the command examples:

– (minus) or / (slash) sign

Flags are prefixed with a – (minus) or / (slash) sign. Flags define the action of a command or modify the operation of a command. You can use multiple flags, followed by parameters, when you issue a command.

[] square brackets

Optional values are enclosed in square brackets.

{ } braces

Required or expected values are enclosed in braces.

| vertical bar

A vertical bar signifies that you choose only one value.

For example, [a | b] indicates that you can choose a, b, or nothing. Similarly, { a | b } indicates that you must choose either a or b.

... ellipsis

An ellipsis signifies the values that can be repeated on the command line.

Emphasis

The following typefaces are used to show emphasis:

boldface

Text in **boldface** represents menu items and command names.

italics

Text in *italics* is used to emphasize a word. In command syntax, it is used for variables for which you supply actual values.

monospace

Text in monospace identifies the data or command instances that you type, samples of command output, examples of program code or

messages from the system, or names of command flags, parameters, arguments, and name-value pairs.

Configuration commands

The following sections describe the following CIM agent configuration commands:

- `setuser`
- `setdevice`

setuser

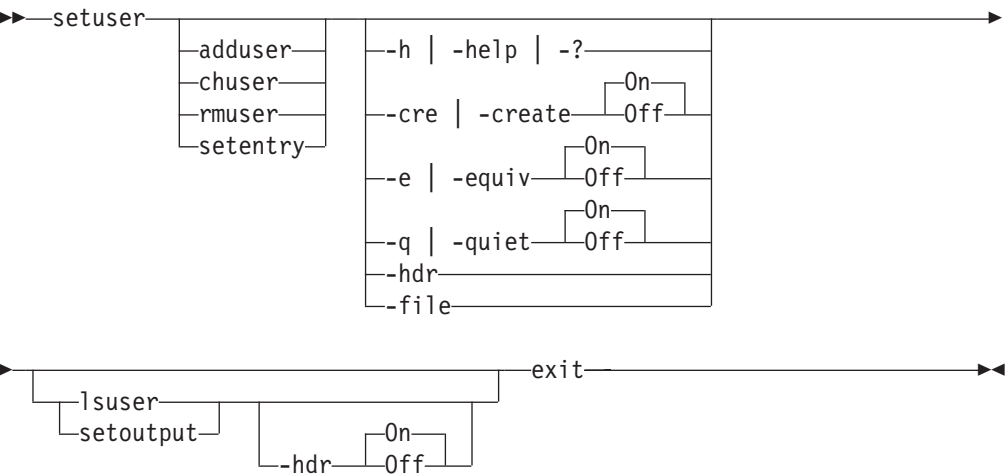
Description

Use the **setuser** command to update the configuration for the users of the ESS CIM Agent. To use the **setuser** command, you must have a username and password to access the ESS CIM Agent.

Syntax

The following is a complete syntax of the **setuser** command.

setuser syntax



Parameters

The following section describes the parameters you can use with the **setuser** command.

adduser

Use the **adduser** command to add information about the user to the configuration file.

chuser

Use the **chuser** command to change the user entry in the configuration file.

lsuser

Use the **lsuser** command to list the users that currently exist in the configuration file.

rmuser

Use the **rmuser** command to remove the user from the configuration file.

setoutput

Use the **setoutput** command to list or set the `-hdr` option globally.

setentry

Use the **setentry** command to list or set the **-create**, **-quiet**, or **-equiv** options globally.

exit

Use the **exit** command to terminate the interactive session.

Command and parameter options: The following section describes the parameter options you can use with the **setuser** command and parameters.

-h | -help | ?

Type **-h | -help | ?** to display the help for a command.

-cre | -create

Type **-cre | -create** to create a password file for the **adduser** and **chuser** commands.

For the **-cre | -create** command, you can set the value to either on or off. If you set the value to on, you can create the password file if it does not already exist. If you set the value to off, you receive an error that the password does not exist.

-e | -equiv

Type **-e | -equiv** to make the **adduser** and **chuser** commands equivalent.

For the **-e | -equiv** command, you can set the value to either on or off. If you set the value to on, you insert the record if a record with the same key does not exist, or to update the existing record. If you set the value to off, the **adduser** and **chuser** commands are not equivalent.

-q | -quiet

Type **-q | -quiet** to prevent the dialog for confirmation or parameter input.

For the **-q | -quiet** command, you can set the value to either on or off. If you set the value to on, you prevent the dialog for confirmation or parameter input. If you set the value to off, you get confirmation for the **chuser** and **rmuser** commands.

-hdr

Type **-hdr** to display the header for the **lsuser** command.

For the **-hdr** command, you can set the value to either on or off. If you set the value to on, you display the header for the **lsuser** command. If you set the value to off, you do not display the header for the **lsuser**.

-file

Type **-file** to specify the password file.

Examples

The following section shows examples and descriptions of the command syntax for the **setuser** command, the parameters, parameter options, and parameter variables.

Figure 38 on page 78 shows how to use the **setuser** command with the **adduser** command to create the password file under the directory where CIMOM resides (if it does not already exist), and inserts an entry for user aaa, where aaa is the username and abc is the password, .

```
setuser
>>>
adduser aaa abc
```

*Figure 38. Example of command syntax to create a password file with the **setuser** and **adduser** commands*

Figure 39 shows how to use the **setuser** command with the **adduser** command to create the password file under the directory where CIMOM resides. Because you did not specify a user or password, you will see commands prompts for Enter user, Enter password, and Confirm password.

```
setuser -e on adduser -q off
```

*Figure 39. Example of command syntax to create a password file with the **setuser** and **adduser** commands without using the parameters for user and password*

Figure 40 shows how to use the **setuser** command with the **lsuser** command to list information about useraaa, without displaying the header, and where aaa is the user.

```
setuser
>>>
lsuser -hdr off aaa
```

Figure 40. Example of the command syntax to display information about the user without displaying a header

Figure 41 shows how to use the **setuser** command with the **lsuser** command to display the help information for the **lsuser** command.

```
setuser lsuser -help
```

*Figure 41. Example of the syntax to display help information for the **lsuser** command*

Figure 42 shows how use the **setuser** command with the **lsuser** command to list all users that exists in the configuration file.

```
setuser lsuser
```

Figure 42. Example of the syntax to display all users in the configuration file

Figure 43 shows how to use the **setuser** command with the **rmuser** command to remove the specified entry from the password file, where aaa is the username.

```
setuser
>>>
rmuser aaa
```

Figure 43. Example of command syntax to remove a user from the password file

Figure 44 on page 79 shows how to use the **setuser** command with the **rmuser** command to remove the specified user from the password file. Before you remove the user, you get a command prompt that asks you whether or not you want to remove the user.

```
setuser -q off rmuser aaa
>>>
Do you want to remove the entry for user aaa (Yes or No)?
```

Figure 44. Example of command syntax to remove a user from the password file

Figure 45 shows how to use the **setuser** command with the **setoutput** command to list or set the header globally.

```
setuser
>>>
setoutput -hdr = on
```

Figure 45. Example of the syntax to display the header globally

Figure 46 shows how use the **setuser** command with the **setoutput** command so that the header does not display.

```
setuser
>>>
setoutput -hdr = off
```

Figure 46. Example of the syntax to not display the header

Figure 47 shows how to use the **setuser** command with the **setentry** command to set the default value globally.

```
setuser
>>>
setentry -equiv = off -quit = on -create = on
```

```
setuser
>>>
setentry -e on
```

```
setuser
>>>
-equiv = on -quit = on -create = on
```

Figure 47. Example of the syntax to set the **-create**, **-quiet**, or **-equiv** options globally

exit: Figure 48 shows how to terminate and exit the interactive session.

```
exit
```

Figure 48. Example of the syntax to terminate and exit the interactive session.

Related topics

- See “startcimom” on page 84 for the **startcimom** command.

setdevice

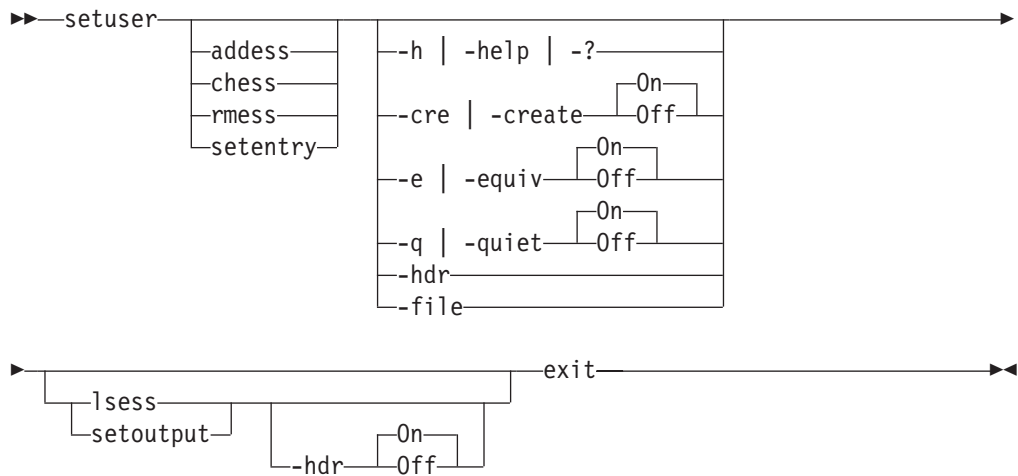
Description

Use the **setdevice** command to update the configuration of managed devices. You can also use the **setdevice** command to define the ESS that is managed by the ESS CIM Agent. You must have a username, password, and valid internet protocol address of an ESS to access the ESS CIM Agent.

Syntax

The following is a complete syntax of the **setdevice** command.

setdevice syntax



Parameters

The following section describes the parameters you can use with the **setdevice** command.

address

Use the **address** command to add the specified ESS provide entry to the configuration file.

chess

Use the **chess** command to change the specified ESS provide entry to the configuration file.

lsess

Use the **lsess** command to list the ESS providers that currently exist in the configuration file and meet the optional search criteria.

rmess

Use the **rmuser** command to remove the specified ESS provider entry from the configuration file.

setoutput

Use the **setoutput** command to list or set the -hdr option globally.

setentry

Use the **setentry** command to list or set the -create, -quiet, or -equiv options globally.

exit

Use the **exit** command to terminate the interactive session.

Command and parameter options: The following section describes the options you can use with the **setdevice** command and parameters.

-h | -help | ?

Type **-h | -help | ?** to display the help for a command.

-cre | -create

Type **-cre | -create** to create a password file for the **address** and **chess** commands.

For the **-cre | -create** command, you can set the value to either on or off. If you set the value to on, you can create the password file if it does not already exist. If you set the value to off, you receive an error that the password does not exist.

-e | -equiv

Type **-e | -equiv** to make the **address** and **chess** commands equivalent.

For the **-e | -equiv** command, you can set the value to either on or off. If you set the value to on, you insert the record if a record with the same key does not exist, or to update the existing record. If you set the value to off, the **address** and **chess** commands are not equivalent.

-q | -quiet

Type **-q | -quiet** to prevent the dialog for confirmation or parameter input.

For the **-q | -quiet** command, you can set the value to either on or off. If you set the value to on, you prevent the dialog for confirmation or parameter input. If you set the value to off, you get confirmation for the **chess** and **rmess** commands.

-hdr

Type **-hdr** to display the header for the **lsuser** command.

For the **-hdr** command, you can set the value to either on or off. If you set the value to on, you display the header for the **lsess** command. If you set the value to off, you do not display the header for the **lsess**.

-file

Type **-file** to specify the password file.

Examples

The following section shows examples and descriptions of the command syntax for the **setdevice** command, parameters, options, and variables.

Figure 49 shows how to use the **setdevice** command with the **address** command to create the xml file provider-cfg.xml under the directory where CIMOM resides (if it does not already exist), and inserts an entry for user aaa, where aaa is the username and abc is the password,

```
setdevice address -e on 1.2.3.4 aaa abc
```

Figure 49. Example of command syntax to create an xml file provider-cfg.xml using the address command

Figure 50 shows how to use the **setdevice** command with the **address** command to create the xml file provider-cfg.xml under the directory where CIMOM resides. Because you did not specify a user or password, you will see commands prompts for Enter user, Enter password, and Confirm password, where aaa is the username and abc is the password,

```
setdevice
>>>
address -q off -e 1.2.3.4
```

Figure 50. Example of command syntax to create an xml file provider-cfg.xml using the address command without using the parameters for user and password

Figure 51 on page 82 shows how to use the **setdevice** command with the **chess** command to create the xml file provider-cfg.xml under the directory where CIMOM

resides (if it does not already exist), and inserts an entry for user aaa, where aaa is the username and abc is the password,

```
setdevice chess -e on 1.2.3.4 aaa abc
```

Figure 51. Example of command syntax to change an xml file provider-cfg.xml with the chess command

Figure 52 shows how to use the **setdevice** command with the **chess** command to create the xml file provider-cfg.xml under the directory where CIMOM resides. Because you did not specify a user or password, you will see commands prompts for Enter user, Enter password, and Confirm password.

```
setdevice
>>>
chess -q off -e 1.2.3.4
```

Figure 52. Example of command syntax to create an xml file provider-cfg.xml using the chess command without using the parameters for user and password

Figure 53 shows how to use the **setdevice** command with the **lsess** command to list all entries in the ESS section.

```
setdevice lsess
```

Figure 53. Example of the command syntax to display all entries in the ESS section

Figure 54 shows how to use the **setdevice** command with the **lsess** command to list all entries in the ESS section.

```
setdevice
>>>
lsess -help
```

Figure 54. Example of the command syntax to display the help for the lsess command

Figure 55 shows how to use the **setdevice** command with the **lsess** command to list an entry for an internet protocol address without a header.

```
setdevice lsess -hdr off 1.2.3.4
```

Figure 55. Example of the command syntax to list the ESS entry for an internet protocol address without a header

Figure 56 shows how to use the **setdevice** command with the **rmess** command to remove the specified ESS provider entry from the xml file provider-cfg.xml (if it doesn't already exist).

```
setdevice rmess -q on 1.2.3.4
```

Figure 56. Example of the command syntax to remove the specified ESS provider entry from the xml file

Figure 57 on page 83 shows how to use the **setdevice** command with the **rmess** command to remove the specified ESS provider entry from the xml file provider-cfg.xml (if it doesn't already exist). Before you remove the specified ESS

provider entry from the xml file provider-cfg.xml, you get a command prompt that asks: Do you want to delete the specified provider entry? (Yes or No)

```
setdevice
>>>
rmess -q off 1.2.3.4
```

Figure 57. Example of the command syntax to remove the specified ESS provider entry from the xml file

Figure 58 shows how to use the **setdevice** command with the **setoutput** command to list or set the header globally.

```
setdevice
>>>
setoutput -hdr = on
```

Figure 58. Example of the syntax to display the header

Figure 59 shows how to use the **setdevice** command with the **setoutput** command so that the header does not display.

```
setdevice
>>>
setoutput -hdr = off
```

Figure 59. Example of the syntax to not display the header

Figure 60 shows how to use the **setdevice** command with the **setentry** command to set the default value globally.

```
setdevice
>>>
setentry -equiv = off -quit = on -create = on
```

```
setdevice
>>>
setentry -e on
```

```
setdevice
>>>
-equiv = on -quit = on -create = on
```

Figure 60. Example of the syntax to set the **-create**, **-quiet**, or **-equiv** options globally

exit: Figure 61 shows an example of how to terminate and exit the interactive session.

```
exit
```

Figure 61. Example of the syntax to terminate and exit the interactive session.

Related topics

Operational commands

The following sections describe the following CIM agent operational commands

- **startcimom**
- **stopcimom**

startcimom

Description

Use the **startcimom** command to run the CIM Agent code. When you use the **startcimom** command, it registers itself with SLP and accepts requests on port 5989.

This command starts the ESS CIM Agent when the ESS CIM Agent is installed. Generally it is installed as a service or part of the system **inittab**. In most cases, there is no reason to start the ESS CIM Agent manually; however, this command will start the ESS CIM Agent if needed.

The certificate used by the ESS CIM Agent must also be made available to each client software product that intends to communicate with the ESS CIM Agent.

Syntax

The following is a complete syntax of the **startcimom** command.

startcimom syntax



Examples

The following section shows examples and descriptions of the command syntax for the **startcimom** command.

Figure 62 shows an example of what is displayed when you use the **startcimom** command.

```
startcimom
```

Figure 62. Example of what is displayed when you use the **startcimom** command

The resulting output You should see the CIMOM you started registered as a service.

example of resulting output

Related topics

- See “mkcertificate” on page 85 for the **mkcertificate** command.
- See “slpd” on page 86 for the **slpd** command.

stopcimom

Description

Use the **stopcimom** command to stop the ESS CIM Agent.

Syntax

The following is a complete syntax of the **startcimom** command.

stopcimom syntax

```

▶▶ stopcimom [-h | help | ?]

```

Parameters

-h | help | ?

Displays the help for the **stopcimom** command.

Related topics

- See “startcimom” on page 84 for the **startcimom** command.

Utility commands

The following sections describe the following CIM agent utility commands

- mkcertificate
- slpd
- verifyconfig

mkcertificate

Description

The **mkcertificate** runs at install time and can be rerun whenever the user feels their security may be compromised. **mkcertificate** creates an X.509 certificate and places it in a certificate store called truststore. The certificate is required by client code that communicates with the ESS CIM Agent using SSL secure communication. If you have installed a product that uses this type of communication with the ESS CIM Agent, you must be sure that the certificate created with the mkcertificate command is available to all clients / software products that communicate with the ESS CIM Agent.

Syntax

The following is a complete syntax of the **mkcertificate** command.

```

▶▶ mkcertificate [-h | -help | ?]

```

Related topics

- See “startcimom” on page 84 for the **startcimom** command.

slpd

Description

The SLP daemon is automatically started by the installation program or at the system reboot, but it can also be started manually if it is not already running. The ESS Agent registers itself with the SLP daemon to enable client code to find the ESS CIM Agent service. You should always keep the SLP daemon running.

The Service Agents (SA) communicate with each other on one subnet of an IP network to find each other's services. If you install products that use the ESS CIM Agent but run on multiple subnets, you should consider designating one of your SLP daemons to act as a Directory Agent (DA). To do this, you change the configuration used by each SLP daemon (slpd.conf) to point at the IP address of the SLP daemon that is to act as the Directory Agent. You can change this so that your SLP daemon acts as a DA instead. To do this, change the *isDA* configuration variable writing slpd.conf to be true. See the documentation for Open SLP at <http://www.openslp.org/Documentation>.

Syntax

The following is a complete syntax of the **slpd** command.



Related topics

- See “startcimom” on page 84 for the **startcimom** command.

verifyconfig

Description

The **verifyconfig** command is used to locate all WBEM services (for example, CIMOMs) in the local network, display them and then call the CIMOMs on the local machine only requesting information about whether the CIMOMs know of any ESS. Locates all WBEM services in the network. These may include ESS CIM Agents or CIM Agents of other hardware devices. It then communicates with the WBEM service (i.e. CIMOM) running on the local machine, if it is found, and attempts to retrieve some basic instance information from the CIMOM running there.

Syntax

The following is a complete syntax of the **verifyconfig** command.



Parameters

Command and parameter options: The following section describes the options you can use with the **verifyconfig** command.

-h | -help | ?

Type **-h | -help | ?** to display the help for the verifyconfig command.

-b Type **-b** to get basic authentication if authentication is disabled.

-u Type **-u** to to display the username.

-p Type **-p** to display the password.

Return values

Return values

An explanation of the return value. The verifyconfig command will fail if SLPD or CIMOM is not running and if one of the ESS it is managing is not accessible.

Possible failures

- A discussion of something that might cause a failure.

Examples

An invocation example Introduce the executed command
verifyconfig

The resulting output Introduce the resulting output
example of resulting output

Related topics

- See “startcimom” on page 84 for the **startcimom** command.
- See “slpd” on page 86 for the **slpd** command.

Appendix A. CIM model component definitions

This section describes the elements, the namespace, and the object name for the CIM model.

Elements:

The CIM model consists of the following elements: schemas, classes, properties, methods, indications, associations, and references. The following list describes each type of element:

Schema	A schema is a group of classes defined to a single namespace. Within the IBM ESS CIM Agent, the schemas that are supported are the ones loaded through the MOF compiler.
Class	A class is the definition of an object within some hierarchy. Classes can have methods and properties and be the target of an association.
Property	A property is a value used to characterize instances of a class.
Method	A method is an implementation of a function on a class.
Indication	An indication is an object representation of an event.
Association	An association is a class that contains two references which define a relationship between two objects.
Reference	The reference defines the role an object plays in an association.
Qualifier	Qualifiers characterize other elements. They provide additional information about classes, associations, indications, methods, methods parameters, instances, properties or references.

Namespace:

Namespaces define the scope over which a CIM schema applies. The only namespace supported by the ESS CIM Agent is root/cimv2. A CIM schema or version is loaded into a namespace when that schema is compiled by the MOF compiler. CIM operations always execute within the context of a namespace. The namespace must be specified within the message sent by the client to the ESS CIM Agent.

Clients cannot create new namespaces. Attempts to do so result in errors.

Object Name:

An object name consists of a namespace path and a model path. The namespace path provides access to the CIM implementation managed by the ESS CIM Agent. The model path provides navigation within the implementation. An example of an Object name is:

```
http://cimom.storage.sanjose.ibm.com/root/CIMV2:CIMDisk.key1=value1, key2=value2
```

where *http://cimom.storage.sanjose.ibm.com/root/CIMV2* is the namespace path and the rest is the model path.

A CIMObjectPath class defines an object name to the ESS CIM Agent.

Related topics:

- “CIM Agent communication concepts” on page 91
- “CIM Agent Communication Methods” on page 92
- “Error codes returned by the CIMOM” on page 102
- “ESS CIM Agent Class Definitions” on page 108
- “ESS class definition schemas” on page 137

Appendix B. CIM Agent communication information

This section describes communication for the CIM Agent. It includes the following information:

- CIM Agent communication concepts
- CIM Agent communication Methods
- Error codes returned by the CIMOM

CIM Agent communication concepts

This section describes the concepts involved in communication between the ESS CIM Agent and the client application.

Client Communication:

A client application communicates with the ESS CIM Agent through Operation Request Messages encoded within XML. The ESS CIM Agent returns responses with Operation Response Messages. Requests and responses are sub elements of the CIM <MESSAGE> element.

A <MESSAGE> sent to the ESS CIM Agent must contain an ID attribute. A response from the ESS CIM Agent returns this value and thereby enables the client to track requests and their responses

The ESS CIM Agent supports simple requests and simple responses. Simple requests are operation request messages that contain the <SIMPLEREQ> XML tag while simple responses are operation response messages that contain the <SIMPLERSP> tag. A client application determines that the ESS CIM Agent only supports simple operation requests and responses by examining the results of running the OPTIONS method.

Intrinsic and Extrinsic Methods:

All operations on the ESS CIM Agent are performed through the running of one or more methods. A method is either an intrinsic method or an extrinsic method. Intrinsic methods are supported by the ESS CIM Agent itself. These methods are included within XML <IMETHODCALL> tags sent in messages to the ESS CIM Agent. Extrinsic methods are defined by the schema supported by the ESS CIM Agent. These methods are included within XML <METHODCALL> tags sent in messages to the ESS CIM Agent.

Client applications can call on the ESS CIM Agent using the methods described in "CIM Agent Communication Methods" on page 92. . These methods fall within certain functional groups which may or may not actually be supported by the ESS CIM Agent.

Related topics:

- Appendix A, "CIM model component definitions", on page 89
- "CIM Agent Communication Methods" on page 92
- "Error codes returned by the CIMOM" on page 102
- "ESS CIM Agent Class Definitions" on page 108
- "ESS class definition schemas" on page 137

CIM Agent Communication Methods

The following sections and tables list the CIM/XML intrinsic and extrinsic communication methods along with their parameters.

Client application calls to these intrinsic methods result in ESS CIM Agent calls to provider code, if the classes or instances referenced in those calls are surfaced by such code.

The ESS CIM Agent returns <IMETHODRESPONSE> or <METHODRESPONSE> elements to the client application when the intrinsic or extrinsic methods are executed. These elements are contained within a <MESSAGERESPONSE> tag.

GetClass:

Table 2 defines the properties of the **GetClass** method, which is used to return a single class from the target namespace.

Table 2. GetClass method parameters

Parameter	Type	Description
ClassName	String	Defines the name of the class to retrieve.
LocalOnly	Boolean	TRUE returns all properties, methods and qualifiers overridden within the definition of the class.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

Return Values: Either a single class or one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

GetInstance:

Table 3 defines the properties of the **GetInstance** method, which is used to return a single instance from the target namespace.

Table 3. GetInstance method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance to retrieve.
LocalOnly	Boolean	TRUE returns all properties and qualifiers overridden within the definition of the class.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

Return Values: Either a single class or one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED

- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

DeleteClass:

The **DeleteClass** method is used to delete a single class from the target namespace.

Note: This operation is not supported. The CIM_ERR_NOT_SUPPORTED is returned to the client application if a request to execute this operation is received.

DeleteInstance:

Table 4 defines the properties of the **DeleteInstance** method, which is used to delete a single instance from the target namespace.

Table 4. DeleteInstance method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance to delete.

Return Values: The named instance is deleted or one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

CreateClass:

The **CreateClass** method is used to create a new class from the target namespace.

Note: This operation is not supported. The CIM_ERR_NOT_SUPPORTED is returned to the client application if a request to execute this operation is received.

CreateInstance:

Table 5 on page 94 defines the properties of the **CreateInstance** method, which is used to create an instance in the target namespace. The instance must not already exist.

Table 5. CreateInstance method parameters

Parameter	Type	Description
Instance	CIMInstance	The instance to be created. The instance must be based on a class already defined in the target namespace.

Return Values: If successful, the specified instance is created. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_ALREADY_EXISTS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

ModifyClass:

The **ModifyClass** method is used to modify an existing class.

Note: This operation is not supported. The CIM_ERR_NOT_SUPPORTED is returned to the client application if a request to execute this operation is received.

ModifyInstance:

Table 6 defines the properties of the **ModifyInstance** method, which is used to modify an existing instance in the target namespace. The instance must already exist.

Table 6. ModifyInstance method parameters

Parameter	Type	Description
Instance	CIMInstance	Defines the modified instance.

Return Values: If successful, the specified instance is updated. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

EnumerateClasses:

Table 7 on page 95 defines the properties of the **EnumerateClasses** method, which is used to return a single class from the target namespace.

Table 7. *EnumerateClasses* method parameters

Parameter	Type	Description
ClassName	String	Defines the name of the class for which subclasses are to be returned. If this field is NULL, it indicates that all base classes within the target namespace should be returned.
DeepInheritance	Boolean	TRUE indicates that all subclasses of the specified class are to be returned. FALSE indicates that only immediate child subclasses are to be returned.
LocalOnly	Boolean	TRUE returns all properties, methods and qualifiers overridden within the definition of the class.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN of the class.

Return Values: If successful, zero or more classes (CIMClass) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

EnumerateClassNames:

Table 8 defines the properties of the **EnumerateClassNames** method, which is used to enumerate the names of subclasses of a class defined within the target namespace.

Table 8. *EnumerateClassNames* method parameters

Parameter	Type	Description
ClassName	String	Defines the name of the class for which subclass names are to be returned. If this field is NULL, it indicates that all base class names within the target namespace should be returned.
DeepInheritance	Boolean	TRUE indicates that all subclass names of the specified class are to be returned. FALSE indicates that only immediate child subclass names are to be returned.

Return Values: If successful, zero or more class names are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

EnumerateInstances:

Table 9 defines the properties of the **EnumerateInstances** method, which is used to enumerate instances of a class defined in the target namespace.

Table 9. *EnumerateInstances method parameters*

Parameter	Type	Description
ClassName	String	Defines the name of the class for which instances are to be returned.
LocalOnly	Boolean	TRUE returns all properties, methods and qualifiers overridden within the definition of the class.
DeepInheritance	Boolean	TRUE indicates that all instances returned must include all properties of the instance, including those added by subclassing. FALSE indicates that only properties defined for the specified class should be returned.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for each instance, its properties, methods or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class within the instance.

Return Values: If successful, zero or more instances (CIMInstance) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

EnumerateInstanceNames:

Table 10 defines the properties of the **EnumerateInstanceNames** method, which is used to enumerate the names of the instances of a class within a target namespace.

Table 10. *EnumerateInstanceNames method parameters*

Parameter	Type	Description
ClassName	String	Defines the name of the class for which instance names are to be returned.

Return Values: If successful, zero or more names of instances are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

ExecQuery:

Table 11 on page 97 defines the properties of the **ExecQuery** method, which is used to execute a query against the target namespace.

Table 11. ExecQuery method parameters

Parameter	Type	Description
QueryLanguage	String	Defines the query language in which the query parameter is expressed.
Query	String	Defines the query to be executed.

Return Values: If successful, zero or more classes (CIMClass) or instances (CIMInstance) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

Associators:

Table 12 defines the properties of the **Associators** method, which is used to enumerate classes or instances that are associated to a particular CIM Object.

Table 12. Associators method parameters

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name that is the source of the association.
AssocClass	String	If not NULL, indicates that all objects returned must be associated to the source object via an instance of this class or one of its subclasses.
ResultClass	String	If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or be this class.
Role	String	If not NULL, indicates that each return object must be associated to the source object via an association in which the source object plays the specified role. The name of the property in the association class that refers to the source object must match the value of this parameter.
ResultRole	String	If not NULL, indicates that each returned object must be associated to the source object via an association in which the return object plays the specified role. That is, the name of the property in the association class that refers to the returned object must match the value of this parameter.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

Return Values: If successful, zero or more classes (CIMClass) or instances (CIMInstance) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

AssociatorNames:

Table 13 defines the properties of the **AssociatorNames** method, which is used to enumerate the names of the classes or instances that are associated with a particular CIM Object.

Table 13. *AssociatorNames method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instances name that is the source of the association.
AssocClass	String	If not NULL, indicates that all object paths returned identify an object that is associated to the source object via an instance of this class or one of its subclasses.
ResultClass	String	If not NULL, indicates that all returned object paths must identify instances of this class or one of its subclasses or be this class.
Role	String	If not NULL, the name of the property in the association class that refers to the source object must match the value of this parameter.
ResultRole	String	If not NULL, the name of the property in the association class that refers to the return object must match the value of this parameter.

Return Values: If successful, zero or more class paths (CIMObjectPath) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

References:

Table 14 defines the properties of the **References** method, which is used to enumerate the association objects that refer to a particular target class or instance.

Table 14. *References method parameters*

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name whose referring objects are to be returned.
ResultClass	String	If not NULL, indicates that all returned objects must be instances of this class or one of its subclasses or be this class.
Role	String	If not NULL, must be a valid property name. Each returned object must refer to the target object via a property whose name matches the value of this parameter.
IncludeQualifiers	Boolean	TRUE returns all qualifiers for the class, its properties, methods, or method parameters. FALSE returns no qualifiers.
IncludeClassOrigin	Boolean	TRUE returns the CLASSORIGIN attribute of the class.

Return Values: If successful, zero or more classes (CIMClass) or instances (CIMInstance) are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER

- CIM_ERR_INVALID_CLASS
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

ReferenceNames:

Table 15 defines the properties of the **ReferenceNames** method, which is used to enumerate the association objects that refer to a particular target class or instance.

Table 15. ReferenceNames method parameters

Parameter	Type	Description
ObjectName	String	Defines the class name or instance name whose referring objects are to be returned.
ResultClass	String	If not NULL, indicates that all returned object paths must be object paths of instances of this class or one of its subclasses or be this class.
Role	String	If not NULL, must be a valid property name. Each returned object must refer to the target object via a property whose name matches the value of this parameter.

Return Values: If successful, the return value specifies the value of the requested property. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

GetProperty:

Table 16 defines the properties of the **GetProperty** method, which is used to retrieve a single property value from an instance in the target namespace.

Table 16. GetProperty method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance.
Property	String	The name of the property whose value is to be returned from the instance.

Return Values: If successful, the return value specifies the value of the requested property. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

SetProperty:

Table 17 defines the properties of the **SetProperty** method, which is used to set a single property value within an instance in the target namespace.

Table 17. SetProperty method parameters

Parameter	Type	Description
InstanceName	String	Defines the name of the instance.
PropertyName	String	The name of the property whose value is to be updated.

Return Values: If successful, the instance is updated. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_INVALID_CLASS
- CIM_ERR_NOT_FOUND
- CIM_ERR_NO_SUCH_PROPERTY
- CIM_ERR_TYPE_MISMATCH
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

GetQualifier:

Table 18 defines the properties of the **GetQualifier** method, which is used to retrieve a single qualifier declaration from the target namespace.

Table 18. GetQualifier method parameters

Parameter	Type	Description
QualifierName	String	Defines the qualifier whose declaration is to be returned.

Return Values: If successful, the value of the qualifier is returned. Otherwise, one of the following return codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

SetQualifier:

Table 19 on page 101 defines the properties of the **SetQualifier** method, which is used to create or update a qualifier declaration in the target namespace.

Table 19. SetQualifier method parameters

Parameter	Type	Description
QualifierDeclaration	Void	Defines the qualifier declaration to be added to the target namespace.

Return Values: If successful, the qualifier is updated in the target namespace. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_NOT_FOUND
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

DeleteQualifier:

The **DeleteQualifier** method is used to delete a single class from the target namespace.

Note: This operation is not supported. The CIM_ERR_NOT_SUPPORTED is returned to the client application if a request to execute this operation is received.

EnumerateQualifiers:

The **EnumerateQualifiers** method is used to enumerate qualifier declarations from the target namespace.

There are no parameters for this method.

Return Values: If successful, zero or more qualifier declarations are returned. Otherwise, one of the following error codes is returned:

- CIM_ERR_ACCESS_DENIED
- CIM_ERR_INVALID_NAMESPACE
- CIM_ERR_INVALID_PARAMETER
- CIM_ERR_FAILED

See Table 21 on page 103 for an explanation of the error codes.

Functional Groups:

Table 20 on page 102 describes the functional groups supported by the IBM ESS CIM Agent. This information is also returned to a client which makes an OPTIONS request of the ESS CIM Agent.

Table 20. Functional Groups for the ESS CIM Agent

Functional Group	Parameters	Supported / Not Supported
Basic Read	<ul style="list-style-type: none"> • GetClass • EnumerateClasses • EnumerateClassNames • GetInstance • EnumerateInstances • EnumerateInstanceNames • GetProperty 	Supported
Basic Write	<ul style="list-style-type: none"> • SetProperty 	Supported
Schema Manipulation	<ul style="list-style-type: none"> • CreateClass • ModifyClass • DeleteClass 	Not Supported
Instance Manipulation	<ul style="list-style-type: none"> • CreateInstance • ModifyInstance • DeleteInstance 	Supported
Association Traversal	<ul style="list-style-type: none"> • Associators • AssociatorNames • References • ReferenceNames 	Supported
Qualifier Declaration	<ul style="list-style-type: none"> • GetQualifier • SetQualifier • DeleteQualifier • EnumerateQualifiers 	Supported
Query Execution	<ul style="list-style-type: none"> • ExecQuery 	Supported

Related topics:

- “CIM Agent communication concepts” on page 91
- Appendix A, “CIM model component definitions”, on page 89
- “Error codes returned by the CIMOM”
- “ESS CIM Agent Class Definitions” on page 108
- “ESS class definition schemas” on page 137

Error codes returned by the CIMOM

The following are possible error codes returned by CIMOM communication methods.

Return Error Codes:

The IBM Object Manager may return status to the client application in two ways:

- Through HTTP status messages or
- Through error codes contained within <METHODRESPONSE> or <IMETHODRESPONSE> XML tags.

Table 21 describes the status codes that IBM CIM Object Manager may return.

Table 21. Return Error Codes for the CIM Object Manager

Symbolic Name	Code	Definition
CIM_ERR_FAILED	1	A general error occurred that is not covered by a more specific error code.
CIM_ERR_ACCESS_DENIED	2	Access to a CIM resource was not available to the client.
CIM_ERR_INVALID_NAMESPACE	3	The target namespace does not exist.
CIM_ERR_INVALID_PARAMETER	4	One or more parameter values passed to the method were invalid.
CIM_ERR_INVALID_CLASS	5	The specified class does not exist.
CIM_ERR_NOT_FOUND	6	The requested object could not be found.
CIM_ERR_NOT_SUPPORTED	7	The requested operation is not supported.
CIM_ERR_CLASS_HAS_CHILDREN	8	The operation cannot be carried out on this class since it has instances.
CIM_ERR_CLASS_HAS_INSTANCES	9	The operation cannot be carried out on this class since it has instances.
CIM_ERR_INVALID_SUPERCLASS	10	The operation cannot be carried out since the specified superclass does not exist.
CIM_ERR_ALREADY_EXISTS	11	The operation cannot be carried out because an object already exists.
CIM_ERR_NO_SUCH_PROPERTY	12	The specified property does not exist.
CIM_ERR_TYPE_MISMATCH	13	The value supplied is incompatible with the type.
CIM_ERR_QUERY_LANGUAGE_NOT_SUPPORTED	14	The query language is not recognized or supported.
CIM_ERR_INVALID_QUERY	15	The query is not valid for the specified query language.
CIM_ERR_METHOD_NOT_AVAILABLE	16	The extrinsic method could not be executed.
CIM_ERR_METHOD_NOT_FOUND	17	The specified extrinsic method does not exist.

Related topics:

- “CIM Agent communication concepts” on page 91
- Appendix A, “CIM model component definitions”, on page 89
- “CIM Agent Communication Methods” on page 92
- “ESS CIM Agent Class Definitions” on page 108
- “ESS class definition schemas” on page 137

Appendix C. ESS CIM Agent Information

The ESS CIM Agent uses the classes described in this section to manage its model of the ESS. Also included is a visual representation of the CIM agent architecture for the ESS. This information will assist you in writing your CIM-based applications. This section includes the following information:

- ESS CIM Agent class definitions quick reference
- ESS CIM Agent class definitions
- ESS class definition schemas

ESS CIM Agent class definitions quick reference

Table 22 provides a quick reference for the ESS CIM agent class definitions.

Table 22. Caption for a columnar data table

IBMTSESS class definition	Description	Derived From	Related section
IBMTSESS_StorageSystem	Represents the entire ESS machine in the configuration.	CIM_ComputerSystem	Table 23 on page 108
IBMTSESS_Host	Represents a host system that is attached to the ESS.	CIM_ComputerSystem	Table 24 on page 110
IBMTSESS_VolumeSpace	Represents an ESS volume group out of which storage volumes can be created.	CIM_StoragePool	Table 25 on page 110
IBMTSESS_Volume	Represents one storage volume for the ESS.	CIM_StorageVolume	Table 26 on page 111
IBMTSESS_FCPort	Describes a single fibre-channel adapter I/O port on the ESS.	CIM_FCPort	Table 27 on page 114
IBMTSESS_Initiator	Describes a single fibre-channel adapter I/O port on a host.	CIM_LogicalDevice	Table 28 on page 115
IBMTSESS_Controller	Represents a port-volume-initiator access relationship.	CIM_Controller	Table 29 on page 116
IBMTSESS_AccessControlInformation	Describes the type of access available to a controller.	CIM_AccessControlInformation	Table 31 on page 118
IBMTSESS_HardwareAccount	Represents a single host initiator (e.g. the World Wide Name (WWN) of a host Fibre-channel adapter port is an example of a host initiator).	CIM_HardwareAccount	Table 32 on page 119
IBMTSESS_StorageConfigurationService	A service provided by the ESS CIM Agent to enable volume creation.	CIM_StorageConfigurationService	Table 33 on page 120
IBMTSESS_AuthorizationService	A service provided by the ESS CIM Agent to enable initiators to access ports and volumes.	CIM_AuthorizationService	Table 35 on page 121

Table 22. Caption for a columnar data table (continued)

IBMTSESS class definition	Description	Derived From	Related section
IBMTSESS_AccountManagementService	A service provided by the ESS CIM Agent for creating and deleting HardwareAccount instances.	CIM_AccountManagementService	Table 37 on page 124
IBMTSESS_Product	Represents the top level description of physical components that make up a product, in this case, the entire ESS.	CIM_Product	Table 39 on page 125
IBMTSESS_AllocatedFromStoragePort	An association that associates a volume with the volumespace from which it is allocated.	CIM_AllocatedFromStoragePort	Table 40 on page 126
IBMTSESS_StorageSystemToVolume	An association that associates a volume with a storage subsystem.	CIM_SystemDevice	Table 41 on page 126
IBMTSESS_StorageSystemToPort	An association that associates a port with a storage subsystem.	CIM_SystemDevice	Table 42 on page 126
IBMTSESS_StorageSystemToController	An association that associates a controller with a storage subsystem.	CIM_SystemDevice	Table 43 on page 126
IBMTSESS_HostToInitiator	An association that associates a host with an initiator.	CIM_SystemDevice	Table 44 on page 127
IBMTSESS_ControllerToVolume	An association that associates a volume with a controller.	CIM_ControlledBy	Table 45 on page 127
IBMTSESS_DeviceConnection	An association between a StorageSystem and a general type of device.	CIM_DeviceConnection	Table 46 on page 128
IBMTSESS_ConcreteIdentityPort	An association that associates two elements representing different aspects of the same underlying entity.	CIM_ConcreteIdentity	Table 47 on page 128
IBMTSESS_ConcreteIdentityInitiator	An association that associates two elements representing different aspects of the same underlying entity.	CIM_ConcreteIdentity	Table 48 on page 128
IBMTSESS_AuthorizationTarget	An association used to apply authorization decisions to specific target resources.	CIM_AuthorizationTarget	Table 49 on page 129
IBMTSESS_AuthorizationSubject	An association used to apply authorization decisions to specific subjects.	CIM_AuthorizationSubject	Table 50 on page 129
IBMTSESS_AuthorizedUse	An association used to provide an AuthorizationService with the AccessControlInformation it needs to do its job.	CIM_AuthorizedUse	Table 51 on page 129

Table 22. Caption for a columnar data table (continued)

IBMTSESS class definition	Description	Derived From	Related section
IBMTSESS_ManagesAccount	Associates the AccountManagement security service to the Accounts for which it is responsible.	CIM_ManagesAccount	Table 52 on page 129
IBMTSESS_HostedService	An association between a System and the StorageConfigurationService.	CIM_HostedService	Table 53 on page 130
IBMTSESS_HostedService2	An association between a System and the AuthorizationService.	CIM_HostedService	Table 54 on page 130
IBMTSESS_HostedService3	An association between a System and the AccountManagementService.	CIM_HostedService	Table 55 on page 130
IBMTSESS_ElementSetting	An association between ManagedElements and the Setting classes defined for them.	CIM_ElementSetting	Table 56 on page 130
IBMTSESS_ElementCapabilities	An association between ManagedElements and their Capabilities.	CIM_ElementCapabilities	Table 57 on page 130
IBMTSESS_HostedStoragePool	An association that establishes that the StoragePool is defined in the context of the System.	CIM_HostedStoragePool	Table 58 on page 131
IBMTSESS_StorageCapabilities	Defines the Capabilities of a Storage Pool. There is one instance for all the ESS machines.	CIM_StorageCapabilities	Table 59 on page 131
IBMTSESS_StorageSetting	A Service Level Objective. There are two instances for all the ESS machines.	CIM_StorageSetting	Table 60 on page 131
IBMTSESS_Chassis	Represents a chassis containing one cluster.	CIM_Chassis	Table 61 on page 132
IBMTSESS_ComputerSystemPackage	An association between a PhysicalPackage and the ComputerSystem.	CIM_ComputerSystemPackage	Table 62 on page 133
IBMTSESS_ExtraCapacitySet	Describes the type of redundancy available in the storage system.	CIM_ExtraCapacitySet	Table 63 on page 133
IBMTSESS_ConcreteIdentityCapSet	An association that associates two elements representing different aspects of the same underlying entity.	CIM_ConcreteIdentity	Table 64 on page 134
IBMTSESS_ProductPhysicalComponent	An association that indicates that the referenced PhysicalElement is acquired as part of a Product.	CIM_ProductPhysicalComponent	Table 65 on page 134

Table 22. Caption for a columnar data table (continued)

IBMTSESS class definition	Description	Derived From	Related section
IBMTSESS_RemoteServiceAccessPoint	Describes access and addressing information for a remote connection that is known to a local network element.	CIM_RemoteServiceAccessPoint	Table 66 on page 134
IBMTSESS_HostedAccessPoint	An association between a Service AccessPoint and the System on which it is provided.	CIM_HostedAccessPoint	Table 67 on page 135
IBMTSESS_StorageProcessorCard	Represents a type of physical container that can be plugged into another Card or HostingBoard, or is itself a HostingBoard or Motherboard in a Chassis. Each ESS has two instances of IBMTSESS_StorageProcessorCard.	CIM_Card	Table 68 on page 135
IBMTSESS_ComputerSystemPackage	An association between a PhysicalPackage and the ComputerSystem.	CIM_ComputerSystemPackage	Table 69 on page 136
IBMTSESS_MemberOfCollection	An aggregation used to establish membership of ManagedElements in a Collection.	CIM_MemberOfCollection	Table 70 on page 136

Related topics:

- “ESS CIM Agent Class Definitions”
- “ESS class definition schemas” on page 137

ESS CIM Agent Class Definitions

The following class definition tables list the properties, as well as their types, qualifiers and descriptions, for each ESS class definition.

Table 23. IBMTSESS_StorageSystem class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
Caption	String	MaxLen(64)	Short textual description.
Description	String		Textual description of the object.
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_StorageSystem will be “Enterprise Storage Server”.
<i>CIM_ManagedSystemElement</i>			
InstallDate	datetime		When the object was installed. A lack of a value does not indicate that the object is not installed.

Table 23. IBMTSESS_StorageSystem class properties (continued)

Property	Type	Qualifier / Parameter	Description
Name	String	MaxLen (256)	Defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.
OperationalStatus[]	uint16		Indicates the current status of the element. Various functional and nonfunctional statuses are defined. "OK" (2), "Error" (6), "In Service" (11), "No Contact" (12). Note: Multiple statuses are allowed for the same object.
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
EnabledStatus	uint16		Indicates whether the element is currently shutting down (value=4), or in an enabled (value=2) or disabled (value=3) state.
OtherEnabledStatus	String		Describes the element's enabled/disabled state when the EnabledStatus property is set to 1 (other).
<i>CIM_System</i>			
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property enables all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_StorageSystem will be "IBMTSESS_StorageSystem".
Name	String	Key, MaxLen(256)	Serves as key of a System instance in an enterprise environment. The value for IBMTSESS_StorageSystem will be ESS's serial number.
NameFormat	String		Identifies how the System name was generated. The value for IBMTSESS_StorageSystem will be "Other".
<i>CIM_ComputerSystem</i>			
OtherIdentifyingInfo	String[]		Captures additional data, beyond System Name information, that could be used to identify a ComputerSystem. The value for IBMTSESS_StorageSystem will be the value of WWN for the ESS.
IdentifyingDescriptions	String[]		Provides explanations and details behind the entries in the OtherIdentifyingInfo array. Note: Each entry of this array is related to the entry in OtherIdentifyingInfo that is located at the same index. The value will be "WWN".
Dedicated	uint16[]	"BlockServer"	Indicates whether the ComputerSystem is a special-purpose System or a general-purpose System. The value for IBMTSESS_StorageSystem will be 15, which means the System is BlockServer.
<i>IBMTSESS_StorageSystem</i>			
model	String		Describes the model number of the ESS.
codelevel	String		Describes the code level of the ESS.

Table 23. IBMTSESS_StorageSystem class properties (continued)

Property	Type	Qualifier / Parameter	Description
cache	String		Describes the memory size of the ESS in gigabytes.
nvs	String		Describes nonvolatile storage of the ESS in megabytes. NVS is used to store a second copy of write data to ensure data integrity.

Table 24. IBMTSESS_Host class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_Host will be "Host" plus its host name.
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_System</i>			
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property enables all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_Host will be "IBMTSESS_Host".
Name	String	Key, MaxLen(256)	Serves as key of a System instance in an enterprise environment. In this case, it saves the host alias name, such as "SUN1".
NameFormat	String		Identifies how the System name was generated. The value for IBMTSESS_Host will be "Other".
<i>CIM_ComputerSystem</i>			
IBMTSESS_Host			
initiator	String	16 hex chars	Describes the initiator ID on Host, such as "10000000C922CEA3".
profile	String		Describes the profile associated with this host type.
ports	String		Describes the ESS IP ports visible to this initiator.

Table 25. IBMTSESS_VolumeSpace class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			

Table 25. IBMTSESS_VolumeSpace class properties (continued)

Property	Type	Qualifier / Parameter	Description
ElementName	String		This property allows each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_VolumeSpace will be "VolumeSpace on ESS" plus the ESS serial number.
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_StoragePool</i>			
InstanceID	String	Key	Identifies a unique instance of StoragePool. The value for IBMTSESS_VolumeSpace will be the volumeID-ESS ID.
TotalAvailableSpace	uint64		Describes the remaining capacity available in megabytes.
IBMTSESS_VolumeSpace			
redundancy	String		Describes the volume type (RAID level).
cap	real32		Describes the capacity of the volumespace in gigabytes.
free	real32		Describes the remaining capacity available in gigabytes.
basedon	String		Describes the disk group or disk that the volumespace is based on.
volumes	uint16		Describes the number of allocated volumes.
lss	String		Describes any volumes allocated in this volume's space that are associated with this logical subsystem.
Status	String		Describes the status of volumespace.
SystemName	String		Describes the scoping System's Name. The value for IBMTSESS_VolumeSpace is the ESS serial number to which this volumespace belongs.

Table 26. IBMTSESS_Volume class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property allows each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_Volume will be "Volume: LSS" + LSSId + "VolumeID" + volumeID.
<i>CIM_ManagedSystemElement</i>			
OperationalStatus	uint16[]		Indicates the current statuses of the element. The value for IBMTSESS_Volume will be 2, which means "OK".
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			

Table 26. IBMTSESS_Volume class properties (continued)

Property	Type	Qualifier / Parameter	Description
EnabledStatus	uint16		Indicates whether the element is currently shutting down, or in an enabled, or disabled state. The value for IBMTSESS_Volume will be 2, which means "Enabled".
RequestedStatus	uint16		Indicates whether the element should be shutdown (value = 4), enabled (2), disabled (3), taken <i>offline</i> (6) or tested (7) at the next opportunity. The value for IBMTSESS_Volume will be 2.
EnabledDefault	uint16		Indicates an administrator's default/startup configuration for an element's Enabled Status. By default, the element is "Enabled" (value=2).
<i>CIM_LogicalDevice</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_Volume will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_Volume will be the ESS serial number to which this volume belongs.
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_Volume will be "IBMTSESS_Volume".
DeviceID	String	Key, MaxLen(64)	Describes an address or other identifying information to uniquely name the LogicalDevice. The value for IBMTSESS_Volume will be the volume ID obtained from esscli, such as "0A05".
OtherIdentifyingInfo	String[]		Describes the volume serial number. The value will be volume's serial number, such as "DBA001".
Controlled	boolean		Provides a quick interface for finding Devices with no ControlledBy associations to Controllers. True indicates that the Device is connected to one or more Ports (via Controllers). False indicates that the Device exists but is not connected to a port.
<i>CIM_StorageExtent</i>			
Access	uint16		Describes whether the media is readable (value=1), writeable (value=2), or both (value=3), write Once (value=4), unknown (value=0). The value for IBMTSESS_Volume is 3.

Table 26. IBMTSESS_Volume class properties (continued)

Property	Type	Qualifier / Parameter	Description
BlockSize	uint64		Size in bytes of the blocks which form this StorageExtent. The value for IBMTSESS_Volume will be 512.
NumberOfBlocks	uint64		Total number of logically contiguous blocks, of size Block Size, which form this Extent. The total size of the Extent can be calculated by multiplying BlockSize by NumberOfBlocks.
ConsumableBlocks	uint64		The maximum number of blocks, of size BlockSize, which are available for consumption when layering StorageExtents using the BasedOn association. The value for IBMTSESS_Volume will be 0.
DataRedundancy	uint64		The number of complete copies of data maintained. The value for IBMTSESS_Volume will be 1.
SpindleRedundancy	uint16		The number of disk spindles that can fail without data loss. The value for IBMTSESS_Volume will be 2.
DeltaReservation	uint16		The current value for Delta reservation. The value for IBMTSESS_Volume will be 100.
NoSinglePointOfFailure	boolean		Indicates whether or not there exists no single point of failure. The value for IBMTSESS_Volume will be true.
ExtentStatus	uint16[]		This information is captured in the VolumeStatus property. The value for IBMTSESS_Volume will be 2, which means "None/Not Applicable".
SequentialAccess	boolean		Boolean set to TRUE if the Storage is sequentially accessed by a MediaAccessDevice. The value for IBMTSESS_Volume will be false.
IsBasedOnUnderlyingRedundancy	boolean		True indicates that the underlying StorageExtents participate in a StorageRedundancyGroup. The value for IBMTSESS_Volume will be true.
<i>CIM_StorageVolume</i>			
IBMTSESS_Volume			
cap	real32		Describes the volume capacity in units of GB or Cylinders.
unit	String		Describes the unit of volume capacity.
lss	String		Describes the logical subsystem identifiers .
vs	String		Describes the volumespace to which the volume belongs.

Table 27. IBMTSESS_FCPort class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property allows each instance to define a user-friendly name in addition to key properties/identity data, and description information. The value for IBMTSESS_FCPort will be "Port" + portID.
<i>CIM_ManagedSystemElement</i>			
OperationalStatus	uint16[]		Indicates the current statuses of the element. The value for IBMTSESS_FCPort will be 2, which means "OK".
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_LogicalDevice</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_FCPort will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_FCPort will be the ESS serial number to which this controller belongs.
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_FCPort will be "IBMTSESS_FCPort".
DeviceID	String	Key, MaxLen(64)	Describes an address or other identifying information to uniquely name the LogicalDevice. In this case, it is the port number. The value for IBMTSESS_FCPort will be the port ID obtained from esscli, such as "04".
<i>CIM_LogicalPort</i>			
<i>CIM_NetworkPort</i>			
PermanentAddress	String		Defines the network address hardcoded into a port. The value of IBMTSESS_FCPort will be the port worldwide name.
PortType	uint16		PortType is defined to force consistent naming of the type property in subclasses and to guarantee unique enum values for all instances of NetworkPort. The value of IBMTSESS_FCPort will be 11.

Table 27. IBMTSESS_FCPort class properties (continued)

Property	Type	Qualifier / Parameter	Description
Speed	uint64		An estimate of the current bandwidth in Bits per Second. The value for IBMTSESS_FCPort will be 16777216 bits/second.
<i>CIM_FCPort</i>			
IBMTSESS_FCPort			
loc	String		Describes the physical location of the port on the ESS, including Bay a, Adapter y and Port z.
topology	String		
wwn	String		Describes the port worldwide name.
type	String		Describes the adapter type.

Table 28. IBMTSESS_Initiator class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property allows each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_Initiator will be "Initiator" + initID + "on Host" + hostname.
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_LogicalDevice</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_Initiator will be "IBMTSESS_Host".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_Initiator will be the host name such as "superturbo_fc0".
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_Initiator will be "IBMTSESS_Initiator".
DeviceID	String	Key, MaxLen(64)	Describes an address or other identifying information to uniquely name the LogicalDevice. The value for IBMTSESS_Initiator will be the initiator ID obtained from esscli, such as "10000000C922CEA3".
IBMTSESS_Initiator			
ports	String		The ESS I/O ports visible to this initiator.

Table 29. IBMTSESS_Controller class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		<p>This property allows each instance to define a user-friendly name in addition to its key properties/identity data, and description information.</p> <p>The value for IBMTSESS_Controller will be "Port" + portID + "Controller".</p>
<i>CIM_ManagedSystemElement</i>			
OperationalStatus	uint16[]		<p>Indicates the current statuses of the element.</p> <p>The value for IBMTSESS_Controller will be 2, which means "OK".</p>
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_LogicalDevice</i>			
SystemCreationClassName	String	Key, MaxLen(256)	<p>Describes the scoping System's CreationClassName.</p> <p>The value for IBMTSESS_Controller will be "IBMTSESS_StorageSystem".</p>
SystemName	String	Key, MaxLen(256)	<p>Describes the scoping System's Name.</p> <p>The value for IBMTSESS_Controller will be the ESS serial number.</p>
CreationClassName	String	Key, MaxLen(256)	<p>Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.</p> <p>The value for IBMTSESS_Controller will be "IBMTSESS_Controller".</p>
DeviceID	String	Key, MaxLen(64)	<p>The DeviceID for Controller with AuthorizationView equal to false will be the ESS serial number. The DeviceID for Controller with AuthorizationView equal to true will be portID-initiatorID.</p>
AccessGranted	boolean		<p>Provides a quick interface for finding Devices with no AuthorizationSubject association to an AccessControlInformation instance; either directly, or via a Controller. True indicates that the Device has granted access to some consumer. False indicates that no access has been granted.</p> <p>The value for IBMTSESS_Controller will be true.</p>
<i>CIM_Controller</i>			
AuthorizationView	boolean		<p>Set to true if the Controller represents an authorization target.</p>

Table 29. IBMTSESS_Controller class properties (continued)

Property	Type	Qualifier / Parameter	Description
DefaultGlobalAccess	boolean		Set to true if the Controller represents a default global access definition rather than an authorization for a specific subject.
PortNumber	uint64		System level port identification number.
ProtocolSupported	uint16		The protocol used by the Controller to access controlled Devices. The value for IBMTSESS_Controller is 10, which means "SCSI Serial Bus Protocol".
<i>CIM_SCSIController</i>			
IBMTSESS_Controller			

Table 30. IBMTSESS_Controller method properties

Method	Type	Qualifier / Parameter	Description
CreateView()	uint32	[IN, OUT(false), Description("Contains representations of references to IBMTSESS_FCPort instances.")] String Ports[], [OUT, IN(false), Description("The new Controller with AuthorizationView set to true.")] IBMTSESS_Controller REF View	Clones a IBMTSESS_Controller with AuthorizationView set true. The new controller is used as an AuthorizationTarget. After the new controller becomes persistent, the provider will create a DeviceConnection association between the original and new Controller, and ConcreteIdentity association to the LogicalPort passed in as a parameter. The following are return value descriptions: 0 Success 1 Not Supported 4 Failed (Result might be intermittent, you might need to look into it.) 5 Invalid Parameter Controller 0x1000 Input controller must have AuthorizationView false 0x8010 Support Single Port only 0x8011 Cannot create temporary controller
DeleteView()	uint32	[IN(false), OUT, Description("The Controller created by CreateView.")] IBMTSESS_Controller REF Controller	Delete the controller and associations from CreateView. The following are return value descriptions: 0 Success 1 Not Supported 4 Failed (Result might be intermittent. You might need to look into it.) 0x8010 Should remove access first 0x8020 Controller processing fails (Fail to delete temporary controller from repository server)

Table 30. IBMTSESS_Controller method properties (continued)

Method	Type	Qualifier / Parameter	Description
AttachDevice()	uint32	[IN, OUT(false), Description("The LogicalDevice instance to attach.")] IBMTSESS_Volume REF Device, [IN, OUT(false), Description("The number assigned to ControlledBy.DeviceNumber.")] String DeviceNumber	Associate a IBMTSESS_Volume to this controller. The following are return code descriptions: 0 Success 1 Not Supported 3 Time out 4 Failed (Result might be intermittent. You might need to look into it.) 5 Not valid Controller 0x1000 Invalid LogicalDevice instance 0x1001 Hardware implementation requires null deviceNumber 0x8000 Authorization failure 0x8010 Should AssignAccess first 0x8011 LogicalDevice instance is already attached to a Host 0x8020 IBMTSESS cannot attach device as specified
RemoveDevice()	uint32	[IN, OUT(false), Description("The LogicalDevice instance to attach.")] IBMTSESS_Volume REF Device	Remove the ControlledBy association subclass between this controller and device. The following are return value descriptions: 0 Success 1 Not Supported 3 Time Out 4 Failed (Result might be intermittent. You might need to look into it.) 5 Not valid Controller 0x1000 Invalid LogicalDevice instance 0x8000 Authorization failure 0x8001 Can Not remove device since it is not attached 0x8020 IBMTSESS cannot remove device as specified

Table 31. IBMTSESS_AccessControllInformation class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_AccessControllInformation is "AccessControllInformation".
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_AccessControllInformation</i>			

Table 31. IBMTSESS_AccessControlInformation class properties (continued)

Property	Type	Qualifier / Parameter	Description
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_AccessControlInformation will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_AccessControlInformation will be the ESS serial number.
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_AccessControlInformation will be "IBMTSESS_AccessControlInformation".
Name	String	Key, MaxLen(64)	Defines the unique label. The value for IBMTSESS_AccessControlInformation will be portID-initiatorID.
IBMTSESS_AccessControlInformation			

Table 32. IBMTSESS_HardwareAccount class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_HardwareAccount will be "HardwareAccount" + initID.
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_Account</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_HardwareAccount will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_HardwareAccount will be the host name such as "Fiona".
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_HardwareAccount will be "IBMTSESS_HardwareAccount".
Name	String	Key, MaxLen(64)	Defines the label by which the object is known. The value for IBMTSESS_HardwareAccount will be initiator's WWN.

Table 32. IBMTSESS_HardwareAccount class properties (continued)

Property	Type	Qualifier / Parameter	Description
<i>CIM_HardwareAccount</i>			
UserIDType	uint16		Describes the type of the UserID property, In this case, it is 2 which represents PortWWN.
IBMTSESS_HardwareAccount			
profile	uint16		Defines the platform or operating system type associated with the host. ValueMap {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15"} Values {"AIX", "DGUX", "DYNIX", "HPUX", "IRIX", "ISCSIGATE", "LINUX", "LINUXPPC", "LINUX390", "NETWARE", "NT4", "OPENVMS", "OS400", "SOLARIS251", "SOLARIS26", "TRU64"}

Table 33. IBMTSESS_StorageConfigurationService class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_StorageConfigurationService will be StorageConfigurationService".
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_Service</i>			
<i>CIM_StorageConfigurationService</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_StorageConfigurationService will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_StorageConfigurationService will be the ESS serial number.
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property enables all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_StorageConfigurationService will be "IBMTSESS_StorageConfigurationService".
Name	String	Key, MaxLen(64)	Describes an address or other identifying information to uniquely name the LogicalElement. The value for IBMTSESS_StorageConfigurationService will be "ConfigService".

Table 33. IBMTSESS_StorageConfigurationService class properties (continued)

Property	Type	Qualifier / Parameter	Description
IBMTSESS_StorageConfigurationService			

Table 34. IBMTSESS_StorageConfigurationService method properties

Method	Type	Qualifier / Parameter	Description
CreateOrModifyElementFromStoragePool()	uint32	[IN, Description ("Enumeration indicating the type of element being created or modified. If the input parameter Element is specified (for the case of modify), this type value must match the type of that instance.") Uint16 ElementType, [OUT, IN(false), Description ("Reference to the job (may be null if job completed).") CIM_ConcreteJob ref Job, [IN, Description ("The requirements for the element to maintain. If set to a null value, the default configuration from the source pool will be used.") CIM_StorageSetting ref Goal, [IN, OUT, Description ("As an input parameter, Size specifies the desired size. As an output parameter, Size specifies the size achieved.") Uint64 Size, [IN, Description ("The Pool to create the Element from. This parameter must not be set to null.") IBMTSESS_VolumeSpace ref InPool, [IN, OUT, Description ("Returned parameter is a reference to the resulting element.") IBMTSESS_Volume ref TheElement	<p>Start a job to create or modify a specified element (i.e. volume) from a StoragePool. The following are return value descriptions:</p> <p>0 Success</p> <p>1 Not Supported</p> <p>3 Time Out</p> <p>4 Failed (Result might be intermittent. You might need to look into it.)</p> <p>0x1001 Input size is bigger than the free spaces left in the VolumeSpace</p> <p>0x8000 Authorization failure</p> <p>0x8002 Invalid StorageSetting</p> <p>0x8004 Input size is invalid, either less than or equal to 0, or is null.</p> <p>0x8005 Invalid VolumeSpace</p> <p>0x8010 ElementType should be 2, meaning "StorageVolume". (The Volume input should be null.)</p> <p>0x8020 IBMTSESS does not support modification of volume.</p> <p>0x8021 IBMTSESS cannot create volume as specified.</p>

Table 35. IBMTSESS_AuthorizationService class properties

Property	Type	Qualifier / Parameter	Description
CIM_ManagedElement			
ElementName	String		<p>This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information.</p> <p>The value for IBMTSESS_AuthorizationService will be "AuthorizationService".</p>
CIM_ManagedSystemElement			
CIM_LogicalElement			
CIM_EnabledLogicalElement			

Table 35. *IBMTSESS_AuthorizationService* class properties (continued)

Property	Type	Qualifier / Parameter	Description
<i>CIM_Service</i>			
SystemCreationClassName	String	Key, MaxLen(256)	Describes the scoping System's CreationClassName. The value for IBMTSESS_AuthorizationService will be "IBMTSESS_StorageSystem".
SystemName	String	Key, MaxLen(256)	Describes the scoping System's Name. The value for IBMTSESS_AuthorizationService will be the ESS serial number.
CreationClassName	String	Key, MaxLen(256)	Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified. The value for IBMTSESS_AuthorizationService will be "IBMTSESS_AuthorizationService".
Name	String	Key, MaxLen(64)	Describes an address or other identifying information to uniquely name the LogicalElement. The value for IBMTSESS_AuthorizationService will be "AuthorizationService".
<i>CIM_SecurityService</i>			
<i>CIM_AuthorizationService</i>			
<i>IBMTSESS_AuthorizationService</i>			

Table 36. IBMTSESS_AuthorizationService method properties

Method	Type	Qualifier / Parameter	Description
AssignAccess()	uint32	[IN] IBMTSESS_HardwareAccount REF Subject, [IN] IBMTSESS_AccessControlInformation REF AccessRights, [IN] IBMTSESS_Controller REF Target	<p>Creates a IBMTSESS_AccessControlInformation, and it is linked via an AuthorizationSubject association to IBMTSESS_HardwareAccount, and via AuthorizationTarget association to Controller, and is associated to AuthorizationService via AuthorizedUse. The following are return value descriptions:</p> <p>0 Success</p> <p>1 Not Supported</p> <p>3 Time Out</p> <p>4 Failed (Result might be intermittent. You might need to look into it.)</p> <p>5 Missing required parameter</p> <p>0x8000 Authorization Failure</p> <p>0x8001 Input HardwareAccount is null, or not HardwareAccount or invalid</p> <p>0x8003 Input Controller is null, or its AuthorizationView is True, or invalid (access already assigned)</p> <p>0x8010 Support Single Target only</p> <p>0x8020 IBMTSESS does not support this target to be reassigned.</p> <p>0x8021 Controller processing fails</p> <p>0x8022 HardwareAccount processing fails</p> <p>0x8030 Creating indication failure</p>
RemoveAccess()	uint32	[IN] IBMTSESS_HardwareAccount REF Subject, [IN] IBMTSESS_AccessControlInformation REF Access, [IN] IBMTSESS_Controller REF Target	<p>Deletes the AuthorizedSubject and AuthorizedTarget associations to AccessControlInformation instances for IBMTSESS_HardwareAccount and IBMTSESS_Controller. The following are return value descriptions:</p> <p>0 Success</p> <p>1 Not Supported</p> <p>3 Time Out</p> <p>4 Failed (Result might be intermittent. You might need to look into it.)</p> <p>0x8000 Authorization failure.</p> <p>0x8001 HardwareAccount is null or invlid</p> <p>0x8003 Controller is null or invalid</p> <p>0x8004 Access not yet assigned</p> <p>0x8005 The specified Subject and Target are not associated</p> <p>0x8010 Support Single Target only</p> <p>0x8011 Should DetachDevice first</p> <p>0x8020 IBMTSESS cannot remove access as specified</p> <p>0x8021 Controller processnig fails</p> <p>0x8022 HardwareAccount processing fails</p> <p>0x8030 Creating indication failure</p>

Table 37. IBMTSESS_AccountManagementService class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		<p>This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information.</p> <p>The value for IBMTSESS_AccountManagementService will be "AccountManagementService".</p>
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_Service</i>			
SystemCreationClassName	String	Key, MaxLen(256)	<p>Describes the scoping System's CreationClassName.</p> <p>The value for IBMTSESS_AccountManagementService will be "IBMTSESS_StorageSystem".</p>
SystemName	String	Key, MaxLen(256)	<p>Describes the scoping System's Name.</p> <p>The value for IBMTSESS_AccountManagementService will be the ESS serial number.</p>
CreationClassName	String	Key, MaxLen(256)	<p>Indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property enables all instances of this class and its subclasses to be uniquely identified.</p> <p>The value for IBMTSESS_AccountManagementService will be "IBMTSESS_AccountManagementService".</p>
Name	String	Key, MaxLen(64)	<p>Describes an address or other identifying information to uniquely name the LogicalElement.</p> <p>The value for IBMTSESS_AccountManagementService will be "AccountManagementService".</p>
<i>CIM_SecurityService</i>			
<i>CIM_AccountManagementService</i>			
IBMTSESS_AccountManagementService			

Table 38. IBMTSESS_AccountManagementService method properties

Method	Type	Qualifier / Parameter	Description
CreateHardwareAccount()	uint32	[IN] String Name, [IN] String UserID, [IN] Uint16 UserIDType, [IN] String ObjectClass[], [IN] String Descriptions[], [IN] String Host[], [IN] String LocalityName[], [IN] String OrganizationName[], [IN] String OU[], [IN] String SeeAlso[], [IN] String UserCertificate[], [IN] String UserPassword[], [IN] uint16 Profile, [OUT] IBMTSESS_HardwareAccount ref Account	Creates a IBMTSESS_HardwareAccount and the association IBMTSESS_ManagesAccount between the service and the new IBMTSESS_HardwareAccount. The following are return value descriptions: 0 Success 1 Not Supported 5 Invalid Parameter 0x8010 UserIDType should be PortWWN 0x8011 UserIDType should be the same as Name, which is PortWWN. 0x8020 HardwareAccount instance already exists or HardwareAccount processing fails. 0x8030 Creating indication failure
DeleteHardwareAccount()	uint32	[IN] IBMTSESS_HardwareAccount REF Account	Deletes a named IBMTSESS_HardwareAccount, and eliminates associations that are no longer needed. The following are return value descriptions: 0 Success 1 Not Supported 5 Invalid Parameter 0x8010 Should remove access first 0x8020 HardwareAccount processing fails 0x8030 Creating indication failure

Table 39. IBMTSESS_Product class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_Product will be "IBM Product" + its version.
<i>CIM_Product</i>			
Name	String	Key	Commonly used product name. The value for IBMTSESS_Product will be "IBM TotalStorage Enterprise Storage System (ESS)".
IdentifyingNumber	String	Key	Product identification such as a serial number on software. The value for IBMTSESS_Product will be the ESS serial number.
Vendor	String	Key	The name of the Product's supplier. The value for IBMTSESS_Product will be "IBM".

Table 39. IBMTSESS_Product class properties (continued)

Property	Type	Qualifier / Parameter	Description
Version	String	Key	Product version information. The value for IBMTSESS_Product will be the codelevel obtained from the output report that is generated by the esscli command, with the "list server" parameter specified.
IBMTSESS_Product			

Table 40. IBMTSESS_AllocatedFromStoragePool class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_AllocatedFromStoragePool</i>			
IBMTSESS_AllocatedFromStoragePool			
Antecedent (override)	IBMTSESS_VolumeSpace	Key	StoragePool
Dependent (override)	IBMTSESS_Volume	Key	Subsidiary element

Table 41. IBMTSESS_StorageSystemToVolume class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			
<i>CIM_SystemComponent</i>			
<i>CIM_SystemDevice</i>			
IBMTSESS_StorageSystemToVolume			
GroupComponent (override)	IBMTSESS_StorageSystem	Key	Parent system in the Association
PartComponent (override)	IBMTSESS_Volume	Key	A component of a System

Table 42. IBMTSESS_StorageSystemToPort class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			
<i>CIM_SystemComponent</i>			
<i>CIM_SystemDevice</i>			
IBMTSESS_StorageSystemToPort			
GroupComponent (override)	IBMTSESS_StorageSystem	Key	Parent system in the Association
PartComponent (override)	IBMTSESS_FCPort	Key	A component of a System

Table 43. IBMTSESS_StorageSystemToController class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			

Table 43. *IBMTSESS_StorageSystemToController* class properties (continued)

Property	Type	Qualifier / Parameter	Description
<i>CIM_SystemComponent</i>			
<i>CIM_SystemDevice</i>			
<i>IBMTSESS_StorageSystemToController</i>			
GroupComponent (override)	IBMTSESS_StorageSystem	Key	Parent system in the Association
PartComponent (override)	IBMTSESS_Controller	Key	A component of a System

Table 44. *IBMTSESS_HostToInitiator* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			
<i>CIM_SystemComponent</i>			
<i>CIM_SystemDevice</i>			
<i>IBMTSESS_HostToInitiator</i>			
GroupComponent (override)	IBMTSESS_Host	Key	Parent system in the Association
PartComponent (override)	IBMTSESS_Initiator	Key	A component of a System

Table 45. *IBMTSESS_ControllerToVolume* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_DeviceConnection</i>			
<i>CIM_ControlledBy</i>			
DeviceNumber	String		Address of the associated Device in context of the antecedent Controller. The value for IBMTSESS_ControllerToVolume will be the volume ID.
AccessMode	uint16		Describes the accessibility of the device through the antecedent controller. ValueMap {"2", "3", "4"}, Values {"ReadWrite", "ReadOnly", "NoAccess"} The value for IBMTSESS_ControllerToVolume will be 2.
AccessPriority	uint16		Describes the priority given to accesses of the device through this controller. The value for IBMTSESS_ControllerToVolume will be 0.

Table 45. *IBMTSESS_ControllerToVolume* class properties (continued)

Property	Type	Qualifier / Parameter	Description
AccessState	uint16		Indicates whether the Controller is actively commanding or accessing the Device (value=1) or not (value=2). Also, the value, "Unknown" (0), can be defined. The value for IBMTSESS_ControllerToVolume will be 2.
<i>CIM_SCSILUN</i>			
<i>IBMTSESS_ControllerToVolume</i>			
Antecedent (override)	IBMTSESS_Controller	Key	Controller
Dependent (override)	IBMTSESS_Volume	Key	Controlled Device

Table 46. *IBMTSESS_DeviceConnection* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_DeviceConnection</i>			
<i>IBMTSESS_DeviceConnection</i>			
Antecedent (override)	IBMTSESS_Controller	Key	Logical device
Dependent (override)	IBMTSESS_Controller	Key	Second logical device connected to the Antecedent device

Table 47. *IBMTSESS_ConcreteIdentityPort* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_LogicalIdentity</i>			
<i>CIM_ConcreteIdentity</i>			
<i>IBMTSESS_ConcreteIdentityPort</i>			
SystemElement (override)	IBMTSESS_FCPort	Key	One aspect of the ManagedElement
SameElement (override)	IBMTSESS_Controller	Key	Another aspect of the ManagedElement

Table 48. *IBMTSESS_ConcreteIdentityInitiator* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_LogicalIdentity</i>			
<i>CIM_ConcreteIdentity</i>			
<i>IBMTSESS_ConcreteIdentityInitiator</i>			
SystemElement (override)	IBMTSESS_Initiator	Key	One aspect of the ManagedElement
SameElement (override)	IBMTSESS_HardwareAccount	Key	Another aspect of the ManagedElement

Table 49. IBMTSESS_AuthorizationTarget class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_AuthorizationTarget</i>			
IBMTSESS_AuthorizationTarget			
Antecedent (override)	IBMTSESS_AccessControlInformation	Key	AccessControlInformation that applies to the target set
Dependent (override)	IBMTSESS_Controller	Key	It is the target set of resources

Table 50. IBMTSESS_AuthorizationSubject class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_AuthorizationSubject</i>			
IBMTSESS_AuthorizationSubject			
Antecedent (override)	IBMTSESS_AccessControlInformation	Key	AccessControlInformation that applies to the subject set
Dependent (override)	IBMTSESS_HardwareAccount	Key	It is ManagedElement that represents users

Table 51. IBMTSESS_AuthorizedUse class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_AuthorizedUse</i>			
IBMTSESS_AuthorizedUse			
Antecedent (override)	IBMTSESS_AccessControlInformation	Key	Access Control Information
Dependent (override)	IBMTSESS_AuthorizationService	Key	AuthorizationService that uses an ACI

Table 52. IBMTSESS_ManagesAccount class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_ManagesAccount</i>			
IBMTSESS_ManagesAccount			
Antecedent (override)	IBMTSESS_AccountManagementService	Key	Antecedent
Dependent (override)	IBMTSESS_HardwareAccount	Key	Dependent

Table 53. IBMTSESS_HostedService class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_HostedService</i>			
IBMTSESS_HostedService			
Antecedent (override)	IBMTSESS_StorageSystem	Key	Hosting System
Dependent (override)	IBMTSESS_StorageConfigurationService	Key	Service hosted on the System

Table 54. IBMTSESS_HostedService2 class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_HostedService</i>			
IBMTSESS_HostedService2			
Antecedent (override)	IBMTSESS_StorageSystem	Key	Hosting System
Dependent (override)	IBMTSESS_AuthorizationService	Key	Service hosted on the System

Table 55. IBMTSESS_HostedService3 class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_HostedService</i>			
IBMTSESS_HostedService3			
Antecedent (override)	IBMTSESS_StorageSystem	Key	Hosting System
Dependent (override)	IBMTSESS_AccountManagementService	Key	Service hosted on the System

Table 56. IBMTSESS_ElementSetting class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ElementSetting</i>			
IBMTSESS_ElementSetting			
Element (override)	IBMTSESS_Volume	Key	ManagedElement
Setting (override)	IBMTSESS_StorageSetting	Key	Setting object associated with the ManagedSystem

Table 57. IBMTSESS_ElementCapabilities class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ElementCapabilities</i>			
IBMTSESS_ElementCapabilities			

Table 57. *IBMTSESS_ElementCapabilities* class properties (continued)

Property	Type	Qualifier / Parameter	Description
ManagedElement (override)	IBMTSESS_VolumeSpace	Key	ManagedElement
Capabilities (override)	IBMTSESS_StorageCapabilities	Key	Capabilities object associated with the element

Table 58. *IBMTSESS_HostedStoragePool* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			
<i>CIM_SystemComponent</i>			
<i>CIM_HostedStoragePool</i>			
<i>IBMTSESS_HostedStoragePool</i>			
GroupComponent (override)	IBMTSESS_StorageSystem	Key	Parent system in the Association
PartComponent (override)	IBMTSESS_VolumeSpace	Key	Component of a System
<i>IBMTSESS_StorageCapabilities</i>			

Table 59. *IBMTSESS_StorageCapabilities* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_StorageCapabilities will be "Storage Capabilities for VolumeSpace".
<i>CIM_Capabilities</i>			
InstanceID	String	Key	Identifies an unique instance of Capabilities. The value for IBMTSESS_StorageCapabilities will be "IBMTSESS".
<i>CIM_StorageCapabilities</i>			
<i>IBMTSESS_StorageCapabilities</i>			

Table 60. *IBMTSESS_StorageSetting* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_StorageSetting will be "Storage Setting for RAID5" or "Storage Setting for RAID10".

Table 60. IBMTSESS_StorageSetting class properties (continued)

Property	Type	Qualifier / Parameter	Description
<i>CIM_SettingData</i>			
InstanceID	String	Key	Identifies an unique instance of Capabilities. The value for IBMTSESS_StorageSetting will be "raid5" or "raid10".
<i>CIM_StorageSetting</i>			
IBMTSESS_StorageSetting			

Table 61. IBMTSESS_Chassis class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_Chassis will be "Chassis".
<i>CIM_ManagedSystemElement</i>			
<i>CIM_PhysicalElement</i>			
Tag	String	Key	Uniquely identifies the Physical Element. The value for IBMTSESS_Chassis will be the ESS serial number.
CreationClassName	String	Key	Indicates the name of the class of the subclass used in the creation of an instance. The value IBMTSESS_Chassis will be "IBMTSESS_Chassis".
<i>CIM_PhysicalPackage</i>			
<i>CIM_PhysicalFrame</i>			
<i>CIM_Chassis</i>			
NumberOfPowerCords	uint16		Indicates the number of power cords which must be connected to the Chassis for all the componentry to operate. The value for IBMTSESS_Chassis will be 2.
CurrentRequiredOrProduced	uint16		The value for IBMTSESS_Chassis will be -1.
HeatGeneration	uint16		Amount of heat generated by the Chassis in BTU/hour. The value for IBMTSESS_Chassis will be -1.
ChassisType	uint16[]		Indicates the type of Chassis. The value for IBMTSESS_Chassis will be 22, which means "Storage Chassis".
TypeDescriptions	String[]		Provides more information on the ChassisTypes array entries. The value for IBMTSESS_Chassis will be "IBM Enterprise System Server".

Table 61. IBMTSESS_Chassis class properties (continued)

Property	Type	Qualifier / Parameter	Description
IBMTSESS_Chassis			

Table 62. IBMTSESS_ComputerSystemPackage class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_SystemPackaging</i>			
<i>CIM_ComputerSystemPackage</i>			
IBMTSESS_ComputerSystemPackage			
Antecedent (override)	IBMTSESS_Chassis	Key	PhysicalPackage that realizes a Unitary ComputerSystem
Dependent (override)	IBMTSESS_StorageSystem	Key	UnitaryComputerSystem

Table 63. IBMTSESS_ExtraCapacitySet class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_ExtraCapacitySet will be "Extra Capacity Set".
<i>CIM_Collection</i>			
<i>CIM_SystemSpecificCollection</i>			
InstanceID	String	Key	Identifies a unique instance of collection that is scoped (contained) by a System. The value for IBMTSESS_ExtraCapacitySet will be IBM +ESS serial number.
<i>CIM_RedundancySet</i>			
<i>CIM_ExtraCapacitySet</i>			
MinNumberNeeded	uint32		Indicates the smallest number of elements that must be operational in order to function. The value for IBMTSESS_ExtraCapacitySet will be 1.
MaxNumberSupported	uint32		Indicates the largest number of elements that can participate in the ExtraCapacitySet. A value of 0 indicates that there is not limit on the number of elements. The value for IBMTSESS_ExtraCapacitySet will be 2.
LoadBalancedSet	boolean		Indicates whether load balancing is supported by the ExtraCapacitySet. The value for IBMTSESS_ExtraCapacitySet will be true.
IBMTSESS_ExtraCapacitySet			

Table 64. IBMTSESS_ConcreteIdentityCapSet class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_LogicalIdentity</i>			
<i>CIM_ConcreteIdentity</i>			
IBMTSESS_ConcreteIdentityCapSet			
SystemElement (override)	IBMTSESS_StorageSystem	Key	One aspect of the ManagedElement
SameElement (override)	IBMTSESS_ExtraCapacitySet	Key	Another aspect of the ManagedElement

Table 65. IBMTSESS_ProductPhysicalComponent class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Component</i>			
<i>CIM_ProductPhysicalComponent</i>			
IBMTSESS_ProductPhysicalComponent			
GroupComponent (override)	IBMTSESS_Product	Key	The Product
PartComponent (override)	IBMTSESS_Chassis	Key	PhysicalElement that is part of the Product

Table 66. IBMTSESS_RemoteServiceAccessPoint class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_RemoteServiceAccessPoint will be "System Management URL".
<i>CIM_ManagedSystemElement</i>			
<i>CIM_LogicalElement</i>			
<i>CIM_EnabledLogicalElement</i>			
<i>CIM_ServiceAccessPoint</i>			
SystemCreationClassName	String	Key	Identifies the scoping System's CreationClassName. The value for IBMTSESS_RemoteServiceAccessPoint is "IBMTSESS_StorageSystem".
SystemName	String	Key	Describes the scoping System's Name. The value for IBMTSESS_RemoteServiceAccessPoint is the ESS ID.
CreationClassName	String	Key	Indicates the name of the class or the subclass used in the creation of an instance. The value for IBMTSESS_RemoteServiceAccessPoint is "IBMTSESS_RemoteServiceAccessPoint".

Table 66. *IBMTSESS_RemoteServiceAccessPoint* class properties (continued)

Property	Type	Qualifier / Parameter	Description
Name	String	Key	Uniquely identifies the ServiceAccessPoint and provides an indication of the functionality that is managed. The value for IBMTSESS_RemoteServiceAccessPoint is "RemoteServiceAccessPoint".
<i>CIM_RemoteServiceAccessPoint</i>			
AccessInfo	String		Access and addressing information for a remote connection. The value for IBMTSESS_RemoteServiceAccessPoint is "http://" + IP address of the ESS.
InfoFormat	uint16		Describes the format and interpretation of the AccessInfo property. The value for IBMTSESS_RemoteServiceAccessPoint is 200.
<i>IBMTSESS_RemoteServiceAccessPoint</i>			

Table 67. *IBMTSESS_HostedAccessPoint* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_HostedAccessPoint</i>			
<i>IBMTSESS_HostedAccessPoint</i>			
Antecedent (override)	IBMTSESS_StorageSystem	Key	Hosting System.
Dependent (override)	IBMTSESS_RemoteServiceAccessPoint	Key	SAPs that are hosted on this System

Table 68. *IBMTSESS_StorageProcessorCard* class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_ManagedElement</i>			
ElementName	String		This property enables each instance to define a user-friendly name in addition to its key properties/identity data, and description information. The value for IBMTSESS_StorageProcessorCard is "Cluster One Processor Card" and "Cluster Two Processor Card".
<i>CIM_ManagedSystemElement</i>			
OperationalStatus	uint16[]		Indicates the current statuses of the element. The value for IBMTSESS_StorageProcessorCard is 2, which means "OK".
<i>CIM_PhysicalElement</i>			

Table 68. IBMTSESS_StorageProcessorCard class properties (continued)

Property	Type	Qualifier / Parameter	Description
Tag	String	Key	Uniquely identifies the Physical Element and serves as the Element's key. The value for IBMTSESS_StorageProcessorCard is the ESS ID + "Cluser One" or "Cluster Two".
CreationClassName	String	Key	Indicates the name of the class or the subclass used in the creation of an instance. The value for IBMTSESS_StorageProcessorCard is "IBMTSESS_StorageProcessorCard".
<i>CIM_PhysicalPackage</i>			
<i>CIM_Card</i>			
IBMTSESS_StorageProcessorCard			

Table 69. IBMTSESS_ComputerSystemPackageCard class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_Dependency</i>			
<i>CIM_SystemPackaging</i>			
<i>CIM_ComputerSystemPackaging</i>			
IBMTSESS_ComputerSystemPackageCard			
Antecedent (override)	IBMTSESS_StorageProcessorCard	Key	PhysicalPackage that realizes a Unitary ComputerSystem.
Dependent (override)	IBMTSESS_StorageSystem	Key	ComputerSystem.

Table 70. IBMTSESS_MemberOfCollection class properties

Property	Type	Qualifier / Parameter	Description
<i>CIM_MemberOfCollection</i>			
IBMTSESS_MemberOfCollection			
Collection (override)	IBMTSESS_ExtraCapacitySet	Key	Collection that aggregates members
Member (override)	IBMTSESS_StorageSystem	Key	Aggregated member of the Collection

Related topics:

- "ESS class definition schemas" on page 137
- "CIM Agent communication concepts" on page 91
- Appendix A, "CIM model component definitions", on page 89
- "CIM Agent Communication Methods" on page 92
- "Error codes returned by the CIMOM" on page 102

ESS class definition schemas

The following section illustrates the architecture for the ESS CIM agent. Individual functional views show the components that are used for a specific function.

Individual functional schemas:

The specific functionality provided by the CIMOM permits configuration, discovery of subsystems and devices (LUNs), LUN creation, LUN deletion, LUN masking and LUN unmasking. Each of the following diagrams shows the class definitions that apply to these functions.

Figure 63 shows the classes and their related associations that represent physical components of the ESS.

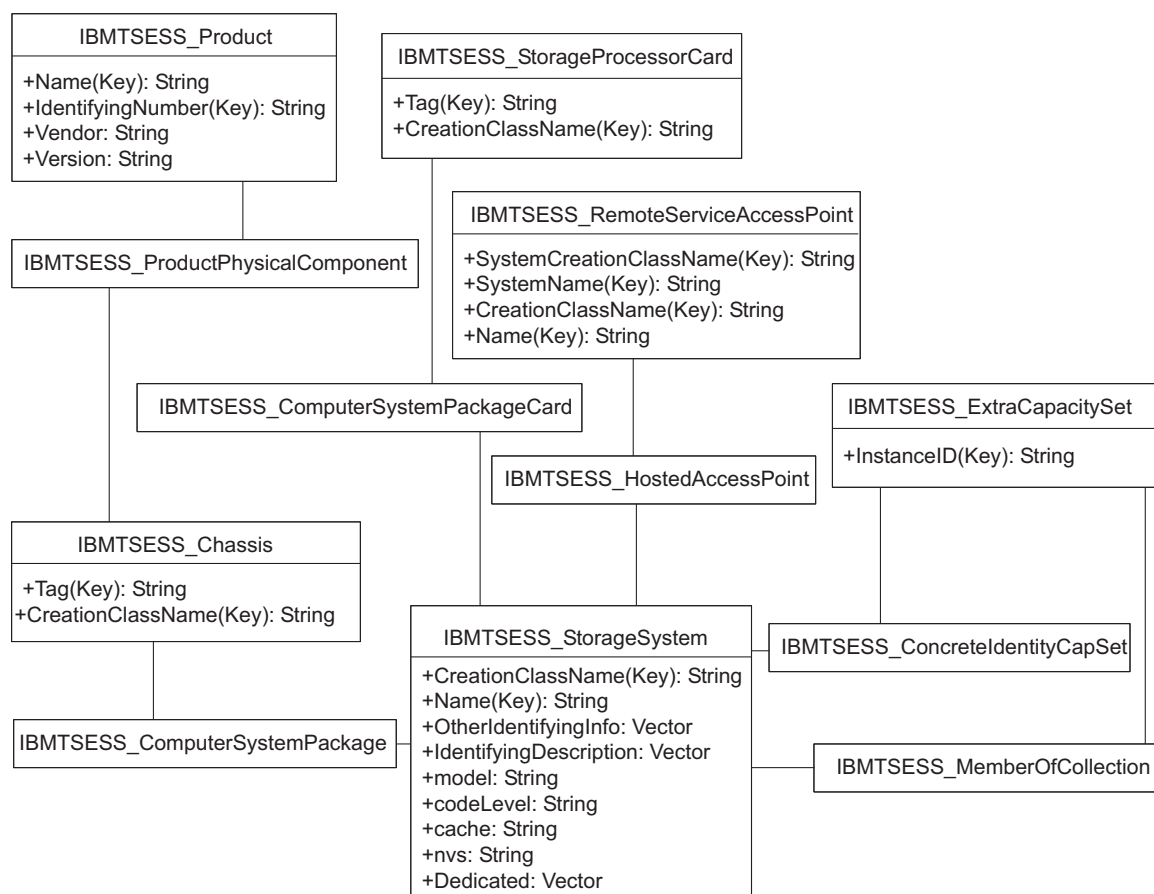


Figure 63. Physical representation

Figure 64 on page 138 shows the classes and their related associations that represent hosts that attach to the ESS.

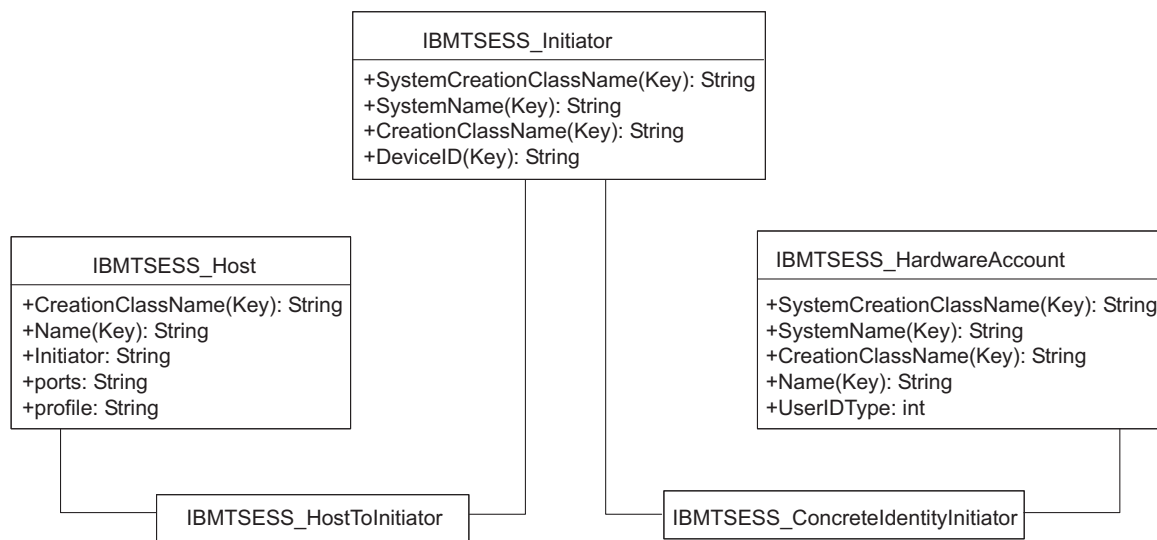


Figure 64. Host representation

Figure 65 shows the classes and their related associations that represent storage pools and volumes.

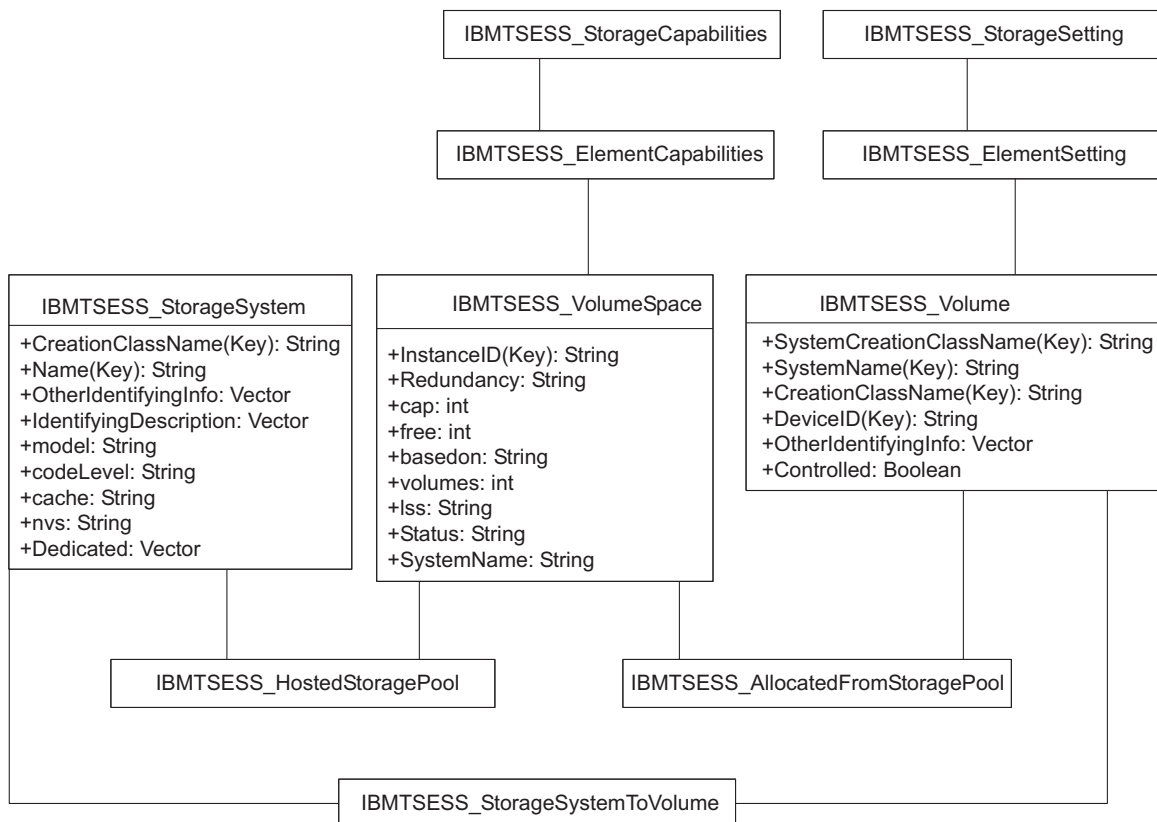


Figure 65. Storage pool and volumes representation

Figure 66 on page 139 shows the classes and their related associations that represent ESS items related to LUN configuration.

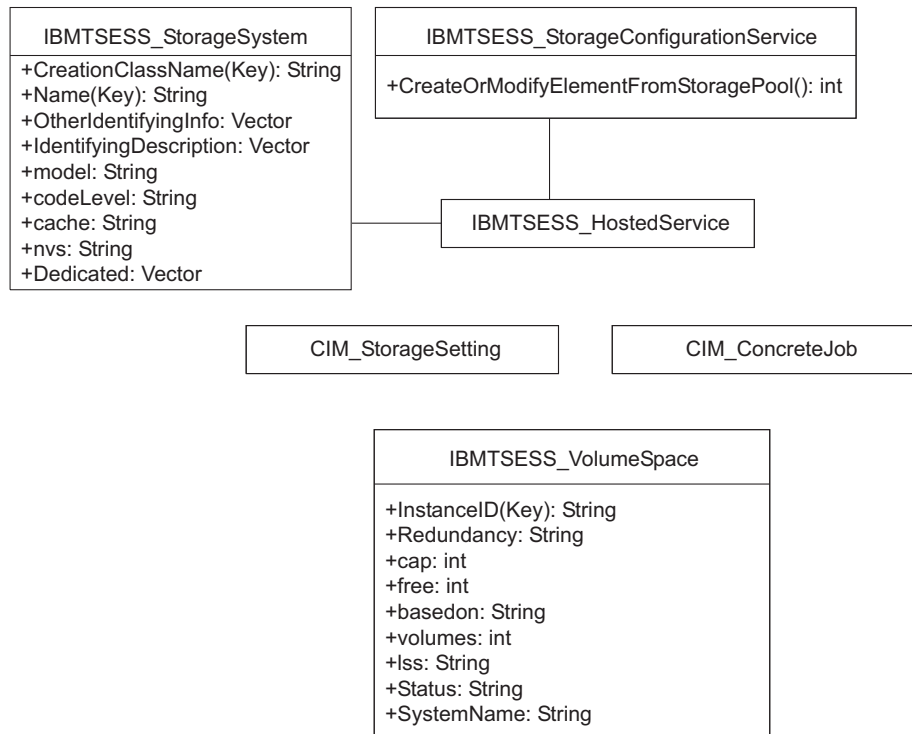


Figure 66. Storage configuration representation

Figure 67 shows the classes and their related associations that represent fibre connectivity from the ESS.

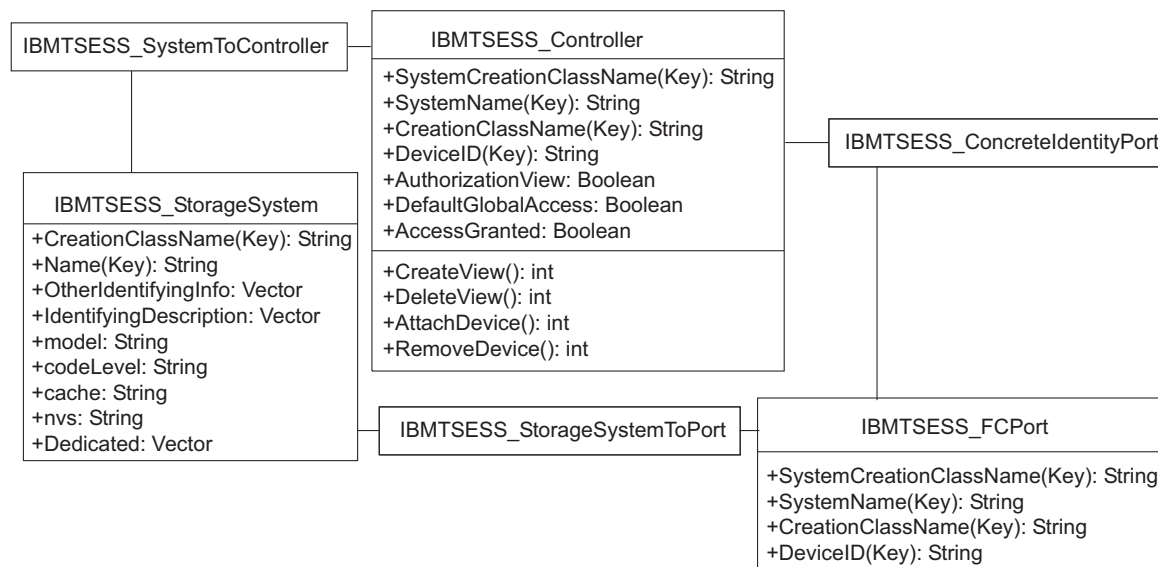


Figure 67. FC transport representation

Figure 68 on page 140 shows the classes and their related associations that represent LUN connectivity to hosts.

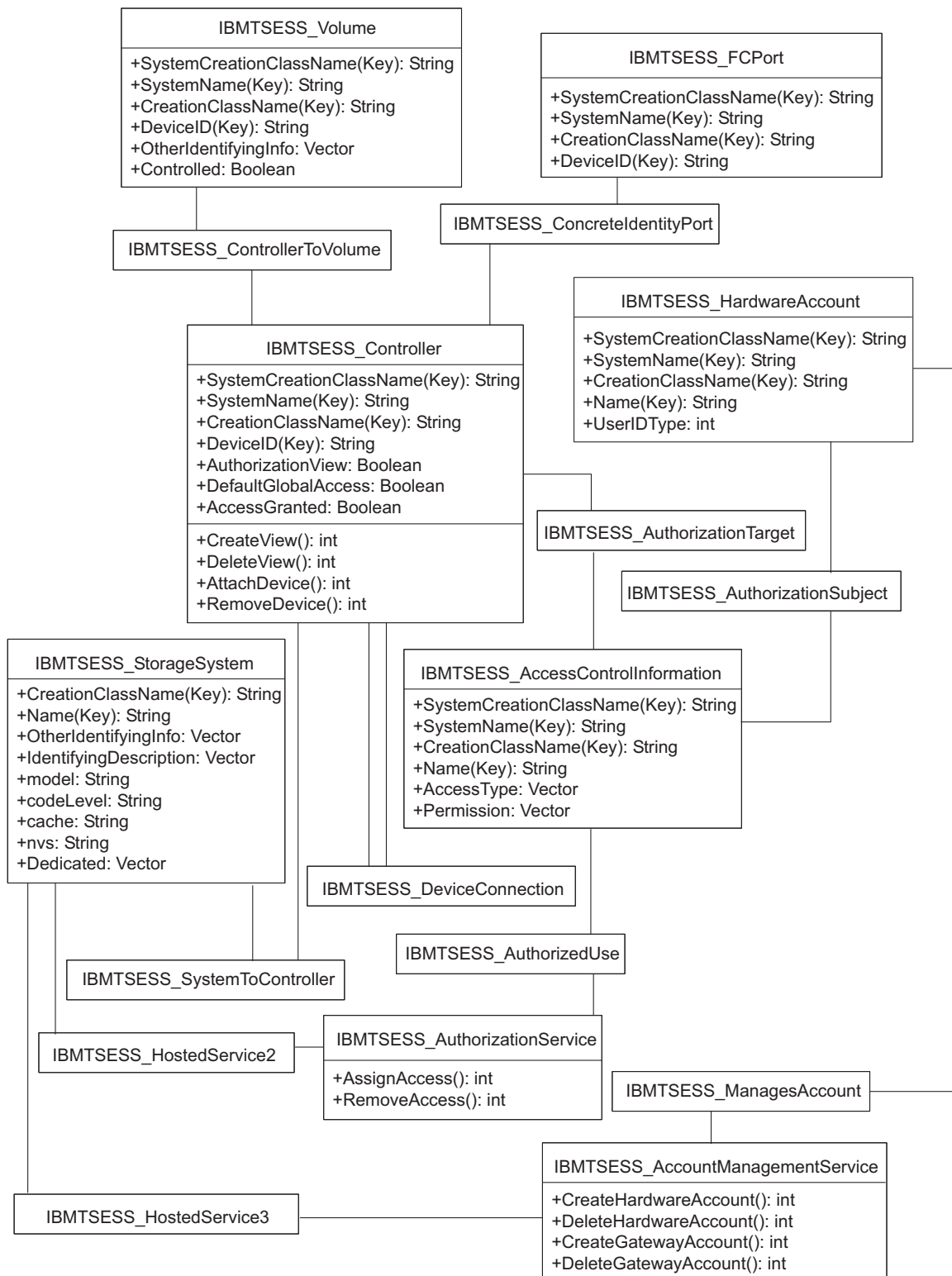


Figure 68. LUN mapping and access representation

Related topics:

- “ESS CIM Agent Class Definitions” on page 108
- “CIM Agent communication concepts” on page 91
- Appendix A, “CIM model component definitions”, on page 89

- "CIM Agent Communication Methods" on page 92
- "Error codes returned by the CIMOM" on page 102

Notices

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

IBM may not offer the products, services, or features that are 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 to cover subject matter that is 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 the following address:

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling the following:

- The exchange of information between independently created programs and other programs (including this one) and
- The mutual use of the information which has been exchanged, should contact:

IBM Corporation
Information Enabling Requests
Dept. DZWA
5600 Cottle Road
San Jose CA 95193-0001
U.S.A.

Such information may be available, subject to appropriate terms and conditions that include 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 following:

- IBM Customer Agreement
- IBM International Program License Agreement
- Or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

Trademarks

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

AIX	InstallShield®
Linux®	TotalStorage
Pentium	Intel®
Enterprise Storage Server	Lotus®
IBM	IBMLink™

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

InstallShield is a registered trademark of the InstallShield Software Corporation.

Lotus® is a trademark of Lotus Development Corporation in the United States, or other countries, or both.

Microsoft®, Windows, and Windows 2000 are registered trademarks of Microsoft Corporation.

Pentium is a registered trademark of the Intel Corporation.

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

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

Java Compatibility Logo

The Java Compatibility Logo identifies products which incorporate a Java Application Environment (JDK or JRE). These products pass the applicable, JavaSoft defined, Java Compatibility test suite in order to enable execution of Java or Personal Java (pJava) applications.



Figure 69. Java Compatibility Logo

Glossary

The terms in this glossary are defined as they pertain to IBM TotalStorage Enterprise Storage Server Expert. If you do not find a term you are looking for, you can refer to the *IBM Dictionary of Computing*, McGraw-Hill, 1994. In the United States and Canada you can order this publication by calling McGraw-Hill at 1-800-2MC-GRAW.

This glossary may include terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright (ANSI). Copies can be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC2/SC1).

A

access.

1. To obtain the use of a computer resource.
2. In computer security, a specific type of interaction between a subject and an object that results in flow of information from one to the other.

access control list.

1. In computer security, a collection of all access rights for one object.
2. In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights; for example, a list associated with a file that identifies users who can access the file and identifies their access rights to that file.

active screen or window. The window or screen a user is currently interacting with.

address.

1. A character or group of characters that identify a register, a particular part of storage, or some other data source or destination.
2. To refer to a device or an item of data by its address.
3. The location, identified by an address code, of a specific section of the recording medium or storage.

4. A name, label, or number identifying a location in storage, a device in a system or network, or any other data source.
5. In data communication, the unique code assigned to each device or workstation connected to a network.

advanced. Usually refers to the performance of functions not understood by the basic user.

advanced interactive executive (AIX). An operating system used in the RISC System/6000 computers. See also "AIX Operating System".

AIX. See "advanced interactive executive (AIX)".

AIX Operating System. IBM's implementation of the UNIX operating system. The RISC 6000 computer, among others, runs on the AIX operating system.

alert. A message or log that a storage facility generates as the result of error event collection and analysis. An alert indicates that you need to perform some service action.

alphanumeric. Pertaining to data that consist of any letters between A through Z, and numbers between 0 through 9.

American National Standard Code for Information Interchange (ASCII). A coding scheme that is defined by ANSI X3.4-1977. Programmers use it to represent various alphabetic, numeric, and special symbols with a seven-bit code.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States.

ANSI. See "American National Standards Institute (ANSI)".

applet. A Java program designed to be run within a client's browser.

applications. A collection of software components used to perform specific types of user oriented work on a computer.

application management. The management of a collection of software components used to perform specific types of user-oriented work on a computer. Application management can improve application availability and performance, increases end-user satisfaction and productivity, lowers IT costs by streamlining problem isolation and resolution, and creates a tighter fit between IT and the business.

ASCII. See “American National Standard Code for Information Interchange (ASCII)” on page 147.

asset management. The organization and arrangement of items, such as storage devices, into useful and logical units. ESRM for example, identifies storage resources by parameters that include, type, model, serial number, features, location, acquisition date, price, and maintenance information.

authentication. In secure communications, a means of verifying the identity of a server or browser (client) with whom you wish to communicate. A sender's authenticity is demonstrated by the digital certificate that is issued to the sender.

authorized personnel. Personnel who are permitted or given authority to communicate with or make use of an object, resource, or function.

automated. The use of automatic procedures to replace or simplify the actions an operator would take in response to system or network events.

availability. For a storage subsystem, the degree to which a data set can be accessed when requested by a user.

B

bit. Binary digit. The storage medium that you need to store a single binary digit.

block. A group of consecutive bytes.

boot. To load an operating system or start the system.

browser. A Web client application for accessing text and images. The Web browser uses an HTTP server across a TCP/IP communication stream.

browser user interface (BUI). A HTTP browser capable of supporting frames, JavaScript and Java applets. This is what the user uses to access Expert via the Internet. In the case of Expert, both Netscape Navigator and Microsoft Internet Explorer are supported.

button.

1. A mechanism on a pointing device, such as a mouse, used to request or initiate an action.
2. A graphical device that identifies a choice.
3. A graphical mechanism in a window that, when selected, results in action; for example, a list button produces a list of choices.
4. A word or picture on the screen that can be selected. Once selected and activated, a button begins an action in the same manner that pressing a key on the keyboard can begin an action.
5. An object that performs an action, e.g., button types include Start, Go, Find, Next, Back, Cancel, Close.

byte. An aggregation of eight bits. The storage medium that you require in order to store eight bits.

C

CA. See “certification authority (CA)”.

CD-ROM. High-capacity read-only memory in the form of an optically read compact disc.

certificate. In electronic commerce, a digital document that binds an encryption key to the identity of the certificate owner, so that the certificate owner can be authenticated. A certificate is issued by a certification authority (CA).

certification authority (CA). In electronic commerce, a trusted third party or a designated internal authority who issues certificates.

CLI. See “command-line interface”.

click. To press and release a button on a mouse or other device without moving the pointer off the item of choice.

client. (1) A function that requests services from a server, and makes them available to the user. (2) An address space in MVS™ that is using TCP/IP services. (3) A term used in an environment to identify a machine that uses the resources of the network.

client authentication. The verification of a client in secure communications whereby the identity of a server or browser (client) with whom you wish to communicate is ascertained. A sender's authenticity is demonstrated by the digital certificate issued to the sender.

client-server relationship. Any process that provides resources to other processes on a network is a server. Any process that employs these resources is a client. A machine can run client processes and server processes at the same time.

close. A choice or command that removes a window and all its associated windows from the screen.

command.

1. A control signal.
2. In a conceptual schema language, the order or trigger for an action or permissible action to take place.
3. Loosely, a mathematical or logical operator.
4. A statement used to request a function of the system.
5. A request from a terminal for the performance of an operation or the execution of a particular program.

command-line interface. The command-line interface on the open-systems hosts that initiates and manages PPRC and FlashCopy on the ESS.

component.

1. Hardware or software that is part of a functional unit.
2. A functional part of an operating system.
3. A set of modules that performs a major function within a system.
4. The widget, gadget, or other graphical object that makes up an interactive user interface.

configuration.

1. The manner in which the hardware and software of an information processing system are organized and interconnected.
2. The physical and logical arrangement of devices and programs that make up a computing system.
3. The devices and programs that make up a system, subsystem, or network.

configuration management. The control of information necessary to identify both physical and logical information system resources and their relationship to one another. ESRM, for example, provides a way to view and modify hardware configurations for host adapters, caches, and storage devices at the licensed internal code, operating system, and software application level.

configure. To define the logical and physical configuration of the input/output (I/O) subsystem via the user interface provided for this function on the storage facility.

D

default. Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified.

device.

1. A mechanical, electrical, or electronic contrivance with a specific purpose.
2. In the AIX operating system, a valuator, button, or the keyboard. Buttons have values of 0 or 1 (up or down); valutors return values in a range, and the keyboard returns ASCII values.
3. The ESA/390 term for the field of an ESCON device-level frame that selects a specific device on a control-unit image.

device adapter (DA). A physical sub-unit of a storage controller that provides the ability to attach to one or more interfaces used to communicate with the associated storage devices.

digital certificate. A file used to identify the authenticity of a person or organization. A certificate is made up of the following:

- The public key of the person who is being certified
- The name and address of the person who is being certified

- The digital signature of the certification authority
- The issue date
- The expiration date

digital signature. A unique mathematically computed signature that ensures accountability.

direct access storage device (DASD). A mass storage medium on which a computer stores data.

disable. To make nonfunctional.

disk drive. A diskette drive or a hard disk drive as opposed to a CD-ROM or optical drive.

disk drive module group (DDM group). A group of eight disk drive modules (DDMs). Each DDM group is a RAID array.

distinguished name. In secure communications, the name and address of the person and organization to whom a certificate has been issued.

distributed system. A system that is spread out across a network, be it a LAN or WAN.

DNS. See "Domain Name System (DNS)".

domain.

1. That part of a computer network in which the data processing resources are under common control.
2. In TCP/IP, the naming system used in hierarchical networks.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

drop-down list. One or more fields which display on the available choices. See also "list box" on page 152.

E

enable. To make functional.

encryption. In secure communications, a means of scrambling data to prevent the data from being read by anyone other than the intended recipient. The sender uses a key to encrypt the message. The recipient uses the decryption key.

Enterprise Storage Server. A disk storage system that provides storage sharing for all major types of servers.

error message. An indication an error has been detected.

ESS. See "Enterprise Storage Server".

event.

1. A representation of a change that occurs to a part. The change enables other interested parts to receive notification when something about the part changes. For example, a push button generates an event by signalling that it has been clicked, which may cause another part to display a window.
2. Any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server.
3. The enqueueing or dequeueing of an element.
4. In computer graphics, information generated either asynchronously from a device or as the side-effect of a client request. Events are grouped into types and are not sent to a client by the server unless the client has issued a specific request for information of that type. Events are usually reported relative to a window.
5. An occurrence of significance to a task or system, such as the completion or failure of an operation.
6. In OSI, the occurrence of a well-defined situation. Events may be planned (for example, transactions), or they may be spontaneous or unplanned (for example, faults). An agent reports events to its managers.
7. A data link control command and response passed between adjacent nodes that allows the two nodes to exchange identification and other information necessary for operation over the data link.

event and alert management. The monitoring of events, receives notification of exception conditions (such as thresholds), initiates exception handlers, and logs information.

event log. A file which lists all actions that have occurred.

extended binary-coded decimal interchange code (EBCDIC). A coding scheme that was developed by IBM, which you use to represent various alphabetic, numeric, and special symbols with an 8 bit code.

F

failed. A status reading indicating the inability of a functional unit to perform its required function.

field.

1. On a data medium or in storage, a specified area used for a particular class of data.
2. The smallest identifiable part of a record.
3. An identifiable area on a screen.

field replaceable unit (FRU). A part that can be replaced in the field.

file. A named set of records stored or processed as a unit.

File Transfer Protocol (FTP). An application protocol used for transferring files to and from host computers. FTP requires a user ID, and a password to allow access to files on a remote host system.

firewall. A protection against unauthorized connection to a computer or a data storage system. The protection is usually in the form of software on a gateway server that grants access to users that meet authorization criteria.

firmware.

1. An ordered set of instructions and data stored in a way that is functionally independent of main storage. Firmware is more efficient than software and is more adaptable to change than pure hardware circuitry.
2. Deprecated term for microcode.

FTP. File Transfer Protocol.

FTP site. An electronic repository of information using File Transfer Protocol (FTP) to transfer data to and from servers.

function.

1. A specific purpose of an entity, or its characteristic action.
2. A system component or licensed program that can be optionally installed in a user's system.

G

GB. See "gigabyte (GB)".

gigabyte (GB). A gigabyte of storage is 109 bytes. A gigabyte of memory is 230 bytes.

graphical user interface (GUI). A type of computer interface consisting of a visual metaphor of a real world scene, frequently a desktop. The GUI contains icons, representing actual objects, that the user can access and manipulate.

H

hard drive. A storage medium within a storage server used to maintain information that the storage server requires.

hdisk. An AIX term for storage space.

help.

1. A standard push button or icon that provides information about the item the cursor is on or about the entire dialog box.

2. An action bar choice that has an associated pull-down. Its pull-down contains choices that can be requested to invoke help actions.

heterogeneous environment. A network where some or all of the computers have dissimilar architecture but nevertheless are able to communicate.

homogeneous. Of the same or similar type. A computer network in which all computers have the same or similar architecture.

host. The controlling or highest level in a data communication configuration. In TCP/IP, any system that has at least one Internet address associated with it. A host with multiple network interfaces may have multiple Internet addresses associated with it.

host application software. Software application programs executed in the host computer.

host initiated. A program executed as a subtask at the request of the host system.

host name. In the Internet suite of protocols, the name given to a computer. Sometimes, "host name" is used to mean fully qualified domain name; other times, it is used to mean the most specific subname of a fully qualified domain name. For example, if mycomputer.city.company.com is the fully qualified domain name, either of the following may be considered the host name:

- mycomputer.city.company.com
- mycomputer

host processor. A processor that controls all or part of a user application network. In a network, the processing unit in which the data communication access method resides. See also "host system".

host system.

1. A computer system that is connected to the ESS. The ESS supports both mainframe (System/390 or zSeries) hosts as well as open-systems hosts. System/390 or zSeries hosts are connected to the ESS through ESCON interfaces. Open-system hosts are connected to the ESS by SCSI or fibre-channel interfaces.
2. The data processing system to which a network is connected and with which the system can communicate.
3. The controlling or highest level system in a data communication configuration.

HP-UX. Hewlett-Packard UNIX operating system.

HTML. See "Hypertext markup language (HTML)".

HTTP. HyperText Transfer Protocol.

Hypertext markup language (HTML). An interpreted markup language used to create hypertext documents.

Hypertext documents can include links to other related documents. HTML controls the format of text and position of form input areas, for example, as well as the navigable links.

Hypertext Transfer Protocol (HTTP). The primary protocol in use on the Web.

I

IBM product engineering (PE). IBM third-level service support IBM engineers who have experience in supporting a product, or who are knowledgeable about the product.

icon.

1. A graphic symbol, displayed on a screen, that a user can point to with a device such as a mouse in order to select a particular function or software application.
2. A graphical representation of an object, consisting of an image, image background, and a label.

ID. See "identifier (ID)".

identifier (ID). A unique name or address that identifies things such as programs, devices, or systems.

initial program load (IPL).

1. The initialization procedure that causes an operating system to commence operation.
2. The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction.
3. The process of loading system programs and preparing a system to run jobs.
4. Synonymous with system restart, and system startup.

initialize. The formatting of a magnetic tape, except for the system files containing information on structure. All former contents of the tape are deleted.

initiate. A network services request sent from a logical unit to a system services control point requesting that a session be established.

initiator. A SCSI term for the part of a host computer that communicates with its attached targets.

insert. To put a tape cartridge in the I/O station or to automatically have the library move a cartridge from the I/O station to a storage slot or drive.

install. Set up for use or service. Adding a product, feature, or function system or device through a simple change or through the addition of multiple components or devices.

instance.

1. In Java programming, when you create a specific variable of a particular class type, it is referred to as instantiating or creating an instance of that class
2. In Java programming, when you create a specific variable of a particular class type, it is referred to as instantiating or creating an instance of that class

interactive.

1. Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system.
2. Pertaining to the exchange of information between a user and a computer.

interchange. The sending and receiving of data in such a manner that the content of the data is not altered during transmission.

interface.

1. A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions.
2. Hardware, software, or both, that links systems, programs or devices.

Internet. A wide area network that connects thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the standard for transmitting information.

Internet Explorer. The Web browser provided by Microsoft Corporation.

Internet Protocol (IP). A protocol used to route data from its source to its destination in the Internet computing network environment.

intranet. A collection of interconnected networks using Internet protocols, but with restricted access and not available to the public.

IP. See "Internet Protocol (IP)".

IP address. A group of four decimal numbers that provides a unique address for the computer.

J

JAR file. A JAR (Java ARchive) file collection of Java classes and other files packaged into a single file. By using a JAR file, the browser makes only one connection to the server rather than several. By reducing the number of files that the browser needs to load from the server, you can download and run your applet that much faster. JAR files can also be compressed, making the overall file size smaller and therefore faster to download.

Java. A programming language that enables application developers to create object-oriented programs that are very secure and portable across different machine and operating system platforms. Java is also dynamic enough to allow for easy expansion.

Java applet. Java code that is compiled into a compact and optimized program.

Java runtime environment (JRE). The underlying, invisible system on your computer that runs applets the browser passes to it.

K

KB. See "kilobyte (KB)".

key ring. In secure communications, a file that contains public keys, private keys, trusted roots, and certificates.

kilobyte (KB). A kilobyte of storage is 103 bytes. A kilobyte of memory is 210 bytes.

L

LAN. See "local area network (LAN)".

link. The connection between two systems or files.

list box. A control that contains scrollable choices from which a user can select one.

local area network (LAN). A computer network located on a user's premises within a limited geographical area.

M

mainframe. A computer, usually in a computer center, with extensive capabilities and resources to which other computers may be connected so that they can share facilities.

MB.

1. For processor storage and real and virtual memory, 1,048,576 bytes.
2. For disk storage capacity and transmission rates, 1,000,000 bytes.

megabyte (MB). See *MB*.

Micro Channel architecture. The rules that define how subsystems and adapters use the Micro Channel bus in a computer. The architecture defines the services that each subsystem can or must provide.

middleware. The term middleware is used to describe separate products that serve as the glue between two applications. It is, therefore, distinct from import and export features that may be built into one of the

applications. Middleware connects two sides of an application and passes data between them.

mid-range systems. A set of multi-use servers with hard disk capacity of 50 GB to 250 GB.

mount.

1. To place a data medium in a position to operate.
2. To make recording media accessible.
3. In AIX, to make a file system accessible.

multiple virtual storage (MVS). Consisting of MVS/System Product Version 1 and MVS/370 Data Facility Product operating on an IBM System/370 processor.

N

Netscape Navigator. The Web browser provided by Netscape Communications Corporation.

network.

1. A configuration of data processing devices and software connected for information interchange.
2. A group of nodes and the links interconnecting them.

network server. The server that controls network operations and management, provides access to files and programs for all the workstations in a particular network.

next. A button or link that allows the user to move forward to the screen which follows the one they are currently viewing.

not installed. A status reading which indicates the device is not present or cannot be initiated at this location.

O

object. An object contains both data and code. The principle feature of an object is that the data is kept privately inside the object structure. Any calculations on that data are also private so it does not matter how that data is stored or how the computations are carried out. It only matters that the object's data and routines are internally consistent and any access to the data in an object always provides consistent results.

offline. The operating condition when the 3584 library cannot interact with host systems.

online. The operating condition when host applications can interact with the 3584 library.

open system. A system whose characteristics comply with standards made available throughout the industry,

and therefore can be connected to other systems that comply with the same standards.

open system hosts.

1. A computer system that is accessed by a user working at a remote location. The term is typically used when two computer systems are connected over a LAN or through a modem.
2. A computer connected to a TCP/IP network, including the Internet. Each host has a unique IP address.
3. (verb) To provide the infrastructure for a computer service.

operating system. The master computer control program that translates user commands and allows software applications to interact with the hardware.

operator. A person or program responsible for managing functions controlled by a given piece of software. The person who operates a device or keeps a system running.

operator intervention. A state requiring the operator of the system or device to take action, either by acknowledging messages and initiating functions or performing problem resolution.

organizationally unique identifier (OUI). An identifier that identifies an organization according to IEEE standards.

P

panel. The formatted display of information that appears on a display screen.

parallel access volume (PAV). Created by associating multiple devices of a single control-unit image with a single logical device. Up to 8 device addresses can be assigned to a parallel access volume.

password. A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it.

path. A statement that indicates where a file is stored on a particular drive. The path consists of all the directories that must be opened to get to a particular file. The directory names are separated by the slash (/).

platform. An ambiguous term that may refer to the hardware, operating system, or a combination of the hardware and the operating system on which software programs run.

platform-independent. Code which is platform-independent can run on multiple combinations of operating systems and hardware.

program. A generic term for the software that controls the operation of a host computer. Typically, the program is an operating system that allows sharing of the host resources between multiple tasks.

protocol. The set of rules that govern the operation of functional units of a communication system. This allows communication to take place. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent. They can also determine high-level exchanges between application programs, such as file transfer.

proxy server. A server that can retrieve documents from other servers for its clients.

public access. Accessible by anyone who has access. If you schedule a task to run and specify public access, all authorized users can view the results. If you do not specify public access, only you have access to the results.

R

RAM. See “random access memory”.

random access memory. A temporary storage location in which the central processing unit (CPU) stores and executes its processes.

register. To record or enroll.

required. A status reading which indicates the current setting is mandatory and cannot be changed.

S

SAN. See “storage area network (SAN)” on page 155.

screen. The physical surface of a display device upon which information is shown to a user.

Secure Sockets Layer (SSL). A protocol that provides secure communications on the Internet. SSL is layered above the connection protocol TCP/IP and beneath application protocols such as HTTP, SMTP, Telnet, FTP, and Gopher.

select.

1. In Client Access for Windows, to highlight a choice so that a subsequent action will use that choice. Selecting does not initiate the action.
2. To choose a button on the display screen.
3. To place the cursor on an object (name or command) and press the Select (left) button on the mouse or the Select key on the keyboard.

server.

1. In a federated database system, a unit of information that identifies a data source to a federated server.

2. A functional unit that provides services to one or more clients over a network.
3. A computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server.
4. A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service.

shared storage. Storage within a storage facility that is configured such that multiple homogenous or divergent hosts can concurrently access the storage. The storage has a uniform appearance to all hosts. The host programs that access the storage must have a common model for the information on a storage device. You need to design the programs to handle the effects of concurrent access.

sign on. A procedure to be followed at a terminal or workstation to establish a link to a computer. (2) To begin a session at a workstation.

silent mode. A program that installs automatically due to specific installation dependent customization information inputted by the user during an umbrella installation.

simple network management protocol (SNMP).

1. An SNMP Manager is part of a network administration software product, such as Tivoli TME 10 Netview.
2. SNMP permits you to define management information base (MIB) extensions, or enterprise-specific MIBs. By loading a MIB file on a manager station, you can monitor the MIB objects. The SNMP Agent accesses MIB objects when a request is received from an SNMP Manager.
3. SNMP is a protocol designed to give a user the capability to remotely manage a computer network by polling and setting terminal values and monitoring network events.

SMIT. See “system management interface tool (SMIT)” on page 155.

software. All or part of the programs, procedures, rules, and associated documentation of a computer processing system. Software is an intellectual creation that is independent of the medium on which it is recorded.

status. The condition or state of hardware or software, usually represented by a status code.

storage. A functional unit into which data or data media can be placed, in which it can be retained and from which it can be retrieved.

storage administrator. A person in the data processing center who is responsible for defining, carrying out, and maintaining storage management policies.

storage area network (SAN). A high speed subnetwork of shared storage devices. A SAN's architecture makes all storage devices available to all servers on a LAN or WAN. As more storage devices are added, they too will be accessible from any server in the larger network. Since stored data does not reside directly on a network's servers, server power is used for applications, and network capacity is released to the end user.

storage device. A physical unit which provides a mechanism to store data on a given medium such that it can be subsequently retrieved.

storage management. The task of using storage, such as DASD, tape, or optical devices, to keep and deliver data to applications.

storage management application. A software tool used to manage storage resources.

storage resources. Data that needs to be managed.

storage subsystem. A storage control and its attached storage devices.

storage server. A unit that manages attached storage devices and provides access to the storage or storage related functions for one or more attached hosts.

submenu. A menu related to and reached through a main menu.

subsystem identification (SSID). A number that uniquely identifies a logical subsystem within a computer installation.

system management interface tool (SMIT). An interface tool of the AIX operating system for installing, maintaining, configuring, and diagnosing tasks.

T

TB. See "terabyte (TB)".

TCP/IP. Transmission Control Protocol/Internet Protocol.

terabyte (TB). A unit of measure equal to 1012 bytes.

terminate, termination.

1. To end a process.
2. The process of ending a process, usually without making any changes.

Transmission Control Protocol/Internet Protocol (TCP/IP).

A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

tutorial. Teaches a user how to use the product. Tutorials generally allow the user to move through the functions at their own pace than reference manuals and usually contain less detail.

U

UA. See "user assistance (UA)".

Uniform Resource Locator (URL). The address convention that indicates the location of an item on the World Wide Web. It includes the protocol followed by the fully-qualified host name, and the request. The server typically maps the request portion of the URL to a path and file name. For example:
<http://www.ibm.com/index.html>.

URL. See "Uniform Resource Locator (URL)".

usage. The number of times a piece of equipment or part has been used by the system, e.g., a cleaning cartridge must be replaced after fifty (50) instances of cleaning a drive.

user assistance (UA). This refers to the help files, where users go to find information to assist them in operations or problem resolution.

User ID. The unique string of characters that identifies any person or device (the User) that may issue or receive commands and messages to or from the information processing system.

V

value.

1. A specific occurrence of an attribute.
2. A quantity assigned to a constant, a variable, parameter or a symbol.

verification. Authentication, certification, or proof, e.g., verification of sensitive data is frequently enforced by dual entry of all the data and a comparison of the results.

W

WAN. Wide area network.

Web. The World Wide Web. The network of HTTP servers that contain programs and files, such as hypertext documents that contain links to other documents on HTTP servers.

wide area network (WAN). A network that provides communication services to a geographic area larger

than served by a local area network or a metropolitan area network, and that may use or provide public communication facilities.

World Wide Web. A global network of servers containing programs and files, accessible by the public.

Special Characters

*** (an asterisk).** This character is used to as a way to include all valid values in a field. For example, if you were specifying IP addresses and used 204.146.18.* you would have specified all valid addresses attached to 204.146.18.

Index

A

about this guide

notational conventions 73

AIX

CLI installation 7

configuring CIM Agent 20

configuring for unsecure mode 21

installation

graphical mode 10

unattended mode 16

installation overview 7

installing the CIM Agent in graphical mode 10

installing the CIM Agent in unattended mode 16

removing the CIM Agent 23

running the CIM Agent 22

verifying installation 19

C

CIM Agent

components 2

configuration on AIX 20

configuration on Linux 42

configuration on Windows 63

configuring for unsecure mode on AIX 21

configuring for unsecure mode on Linux 44

configuring for unsecure mode on Windows 64

installation overview for AIX 7

installation overview for Linux 29

installation overview for Windows 51

installation requirements 4

installing CLI on AIX 7

installing CLI on Linux 29

installing CLI on Windows 51

installing on AIX in graphical mode 10

installing on AIX in unattended mode 16

installing on Linux in graphical mode 31

installing on Linux in unattended mode 38

installing on Windows in graphical mode 53

installing on Windows in unattended mode 60

overview 1, 5

product overview 1

removing from AIX 23

removing from Linux 46

removing from Windows 66

running on AIX 22

running on Linux 44

security 5

verifying connection to the ESS 65

verifying installation on AIX 19

verifying installation on Linux 42

verifying installation on Windows 63

CIM API communication methods

AssociatorNames 98

Associators 97

CreateInstance 93

DeleteInstance 93

CIM API communication methods *(continued)*

EnumerateClasses 94

EnumerateClassNames 95

EnumerateInstanceNames 96

EnumerateInstances 95

EnumerateQualifiers 101

error codes 102

ExecQuery 96

FunctionalGroups 101

GetClass 92

GetInstance 92

GetProperty 99

GetQualifier 100

ModifyInstance 94

ReferenceNames 99

References 98

SetProperty 100

SetQualifier 100

CIM component definitions

core classes 89

device to CIM mapping 89

elements 89

namespace 89

object name 89

CIM overview 3

CIMOM operations

client communication 91

commands

mkcertificate 85

setdevice 79

setuser 76

slpd 86

startcimom 84

stopcimom 85

verifyconfig 86

common CIMOM operations

intrinsic and extrinsic methods 91

configuration

AIX 20

Linux 42

unsecure mode on AIX 21

unsecure mode on Linux 44

unsecure mode on Windows 64

Windows 63

E

emphasis 75

ESS class definitions

individual functional schemas 137

quick reference table 105

I

IBMTSESS_ class definitions

AccessControlInformation 118

AccountManagementService 124

IBMTSESS_ class definitions *(continued)*

- AllocatedFromStoragePool 126
- AuthorizationService 121
- AuthorizationSubject 129
- AuthorizationTarget 129
- AuthorizedUse 129
- Chassis 132
- ComputerSystemPackage 133
- ComputerSystemPackageCard 136
- ConcretIdentityCapSet 134
- ConcretIdentityInitiator 128
- ConcretIdentityPort 128
- Controller 116
- ControllerToVolume 127
- DeviceConnection 128
- ElementCapabilities 130
- ElementSetting 130
- ExtraCapacitySet 133
- FCPort 114
- HardwareAccount 119
- Host 110
- HostedAccessPoint 135
- HostedService 130
- HostedService2 130
- HostedService3 130
- HostedStoragePool 131
- HostToInitiator 127
- Initiator 115
- ManagesAccount 129
- MemberOfCollection 136
- Product 125
- ProductPhysicalComponent 134
- RemoteServiceAccessPoint 134
- StorageCapabilities 131
- StorageConfigurationService 120
- StorageProcessorCard 135
- StorageSetting 131
- StorageSystem 108
- StorageSystemToController 126
- StorageSystemToPort 126
- StorageSystemToVolume 126
- Volume 111
- VolumeSpace 110

installation

- AIX
 - graphical mode 10
 - unattended mode 16
- CLI on AIX 7
- CLI on Linux 29
- CLI on Windows 51
- graphical mode on AIX 10
- graphical mode on Linux 31
- graphical mode on Windows 53
- Linux
 - graphical mode 31
 - overview for AIX 7
 - overview for Linux 29
 - overview for Windows 51
 - unattended mode on AIX 16
 - unattended mode on Linux 38
 - unattended mode on Windows 60

installation *(continued)*

- verifying on AIX 19
- verifying on Linux 42
- verifying on Windows 63
- Windows
 - prerequisites 4

L

Linux

- CLI installation 29
- configuring CIM Agent 42
- configuring for unsecure mode 44
- installation
 - graphical mode 31
- installation overview 29
- installing the CIM Agent in graphical mode 31
- installing the CIM Agent in unattended mode 38
- removing the CIM Agent 46
- running the CIM Agent 44
- verifying installation 42

M

- mkcertificate command 85

N

notational conventions

- emphasis 75
- special characters 75

P

prerequisites

- Windows 4

R

removing the CIM Agent

- AIX 23
- Linux 46
- Windows 66

running the CIM Agent

- on AIX 22
- on Linux 44

S

- setdevice command 79
- setuser command 76
- slpd command 86
- special characters 75
- startcimom command 84
- stopcimom command 85

V

- verifyconfig command 86

- verifying connection to the ESS
 - on Windows 65
- verifying installation
 - AIX 19
 - Linux 42
 - Windows 63

W

- Windows
 - CLI installation 51
 - configuring CIM Agent 63
 - configuring for unsecure mode 64
 - installation overview 51
 - installing the CIM Agent in graphical mode 53
 - installing the CIM Agent in unattended mode 60
 - prerequisites 4
 - removing the CIM Agent 66
 - verifying connection to the ESS 65
 - verifying installation 63



IBM Confidential
Printed in U.S.A.

GC35-0485-00



Spine information:



IBM TotalStorage

CIM Agent User's Guide, Draft 12/10/02

for the IBM Enterprise
Storage Server