Solaris WBEM Services Administration Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 806–6827–10
May 2002

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

CETTE PUBLICATION EST FOURNIE “EN L’ETAT” ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N’EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L’APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU’ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S’APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

Adobe PostScript

020122@3062

Contents

Tables

Figures

Preface

The *Solaris WBEM Services Administration Guide* explains Common Information Model (CIM) concepts and describes how to administer Web-Based Enterprise Management (WBEM) services in the Solaris™ operating environment.

Solaris WBEM Services software makes it easier for software developers to create management applications that run on Solaris and makes the Solaris operating environment easier to manage.

Who Should Use This Book

This book is intended for system administrators who manage WBEM-enabled networks and systems, by running existing WBEM applications or writing new applications.

Before You Read This Book

This book requires knowledge of these topics:

- Object-oriented programming concepts
- Java™ programming
- WBEM Common Information Model (CIM) concepts
- Network management concepts
- Simple Network Management Protocol (SNMP) concepts, if you intend to configure and use the SNMP Adapter for WBEM

If you are unfamiliar with these topics, you might find the following references useful:

- *Java How to Program*, H. M. Deitel and P. J. Deitel, Prentice Hall, ISBN 0–13–263401–5.
- *The Java Class Libraries, Second Edition, Volume 1*, Patrick Chan, Rosanna Lee, Douglas Kramer, Addison-Wesley, ISBN 0–201–31002–3.
- *CIM Tutorial*, provided by the Distributed Management Task Force at `http://www.dmtf.org/education/cimtutorial.php`.

The following web sites are useful resources when working with WBEM technologies:

- **CIM Tutorial Glossary –** `www.dmtf.org/education/cimtutorial/reference/glossary.php`
- **Distributed Management Task Force (DMTF) –** `www.dmtf.org`

  This site discusses the latest developments about CIM, provides information about various working groups, and lists contact information about extending the CIM Schema.
- **Rational Software –** `www.rational.com/uml`

  This site contains documentation about the Unified Modeling Language (UML) and the Rose CASE tool.

How This Book Is Organized

Chapter 1 provides an overview of Web-Based Enterprise Management (WBEM) and Solaris WBEM Services.

Chapter 2 describes the CIM Object Manager. This chapter covers how to start and how to stop the CIM Object Manager and how to upgrade the CIM Object Manager Repository.

Chapter 3 describes SNMP Services for WBEM, which includes the SNMP Adapter for WBEM and the SNMP Provider. Intended for use by system administrators, the SNMP Adapter for WBEM enables Simple Network Management Protocol (SNMP) management applications to access system management information that is provided by Solaris WBEM Services. The SNMP Provider is a software component that provides information about managed elements to the CIM Object Manager, including configuration information about an SNMP device.

Chapter 4 describes WBEM security mechanisms, security features, and how to set access rights for namespaces and users.

Chapter 5 describes how to view log data.

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

CHAPTER 1

WBEM and Solaris WBEM Services (Overview)

This chapter provides an overview of Web-Based Enterprise Management (WBEM) and Solaris WBEM Services. These services make it easier for software developers to create management applications that run on Solaris, and make the Solaris operating environment easier for system administrators to manage.

Here is a list of the information in this chapter.

- "About Web-Based Enterprise Management" on page 15
- "About the Common Information Model" on page 16
- "Solaris WBEM Services" on page 18
- "Solaris WBEM Software Developer's Kit" on page 24

About Web-Based Enterprise Management

WBEM is an industry-wide initiative that includes standards for web-based management of systems, networks, and devices on multiple platforms. This standardization enables system administrators to manage desktops, devices, and networks.

At this time, WBEM is designed to be compatible with the Simple Network Management Protocol (SNMP).

WBEM encompasses the following standards:

- **Common Information Model (CIM) –** Information model for describing managed resources.
- **Managed Object Format (MOF) –** Language for defining CIM classes and instances.

- **eXtensible Markup Language (XML) –** Markup language for describing managed resources on the web.

The Distributed Management Task Force (DMTF), a group that represents corporations in the computer and telecommunications industries, is leading the effort to develop management standards. The goal of the DMTF is to develop an integrated approach to managing networks across platforms and protocols, and consequently promote cost-effective products that interoperate as flawlessly as possible.

About the Common Information Model

This section provides a brief introduction to basic CIM terms and concepts as they are used in the Solaris WBEM Services product. A complete glossary of CIM terms and concepts is provided at `http://www.dmtf.org/education/cimtutorial/reference/glossary.php`.

CIM is an object-oriented information model for describing managed resources such as disks, CPUs, and operating systems. A CIM object is a representation, or model, of a managed resource, such as a printer, disk drive, or CPU. CIM objects can be shared by any WBEM-enabled system, device, or application.

Basic CIM Elements

CIM objects with similar properties and purposes are represented as CIM classes. Properties are attributes that describe a unit of data for a class. An instance is a representation of a managed object that belongs to a particular class. Instances contain actual data. For example, `Solaris_ComputerSystem` is a CIM class that represents a computer that runs the Solaris operating environment. The Solaris software that runs on your system is an instance of the `Solaris_OperatingSystem` class. `ResetCapability` and `InstallDate` are examples of properties of the `Solaris_ComputerSystem` class.

CIM classes are grouped into meaningful collections called schemas. A schema is a group of classes with a single owner (an organization). A class must belong to only one schema. Schemas are used for administration and class naming. All class names must be unique within a particular schema. The schema name is the determining factor in differentiating classes and properties from others that may have the same name. The naming of schema, class, and property follow this syntax:

Schemaname_classname.propertyname

CIM Models

The Common Information Model categorizes information from general to specific. Specific information, such as a representation of the Solaris environment, extends the model. CIM consists of the following three layers of information:

- **Core Model –** Subset of CIM not specific to any platform.
- **Common Model –** Information model that visually depicts concepts, functionality, and representations of entities related to specific areas of network management, such as systems, devices, and applications.
- **Extensions –** Information models that support the CIM Schema and represent a very specific platform, protocol, or corporate brand.

Collectively, the Core Model and the Common Model are called the CIM Schema.

Core Model

The Core Model provides the underlying, general assumptions of the managed environment. For example, specific, requested data must be contained in a location and distributed to requesting applications or users. These assumptions are conveyed as a set of classes and associations that conceptually form the basis of the managed environment. The Core Model is meant to introduce uniformity across schemas that represent specific aspects of the managed environment.

For application developers, the Core Model provides a set of classes, associations, and properties that can be used as a starting point to describe managed systems and determine how to extend the Common Model. The Core Model establishes a conceptual framework for modeling the rest of the managed environment.

The Core Model provides classes and associations to extend specific information about systems, applications, networks, devices, and other network features through the Common Model and extensions.

Common Model

Areas of network management depicted in the Common Model are independent of a specific technology or implementation but provide the basis for the development of management applications. This model provides a set of base classes for extension into the area of five designated technology-specific schemas, that is, Systems, Devices, Applications, Networks, and Physical.

CIM Extensions

Extension schemas are built upon CIM to connect specific technologies to the model. By extending CIM, a specific operating environment such as Solaris can be made available to a greater number of users and administrators. Extension schemas provide classes for software developers to build applications that manage and administer the extended technology. The Solaris Schema is an extension of the CIM Schema.

Solaris WBEM Services

Solaris WBEM Services software provides WBEM services in the Solaris operating environment, including secure access and manipulation of management data. The product includes a Solaris provider that enables management applications to access information about managed resources (devices and software) in the Solaris operating environment.

The CIM Object Manager accepts connections from management applications that use either the Remote Method Invocation (RMI) protocol or the XML/HTTP protocol, and provides the following services to connected clients:

- **Management services –** Are in the form of a CIM Object Manager that checks the semantics and syntax of CIM data and distributes data between applications, the CIM Object Manager Repository, and managed resources.
- **Security services –** Specify these services for WBEM through the Solaris Management Console User tool. These services are described in *System Administration Guide: Security Services*.
- **Sun™ WBEM User Manager –** Use this tool to establish an access control list (ACL) for a specific namespace on the WBEM server. Sun WBEM User Manager enables you to add and delete authorized users, set access privileges for authorized users, and manage user authentication and access to CIM objects on a WBEM-enabled system. ACL-based security is uniquely provided by Solaris WBEM Services.
- **Logging services –** Consist of classes that developers can use to create applications that dynamically record and retrieve event data. Administrators use this data to track and determine the cause of events. Logging services are described in more detail in the *Solaris WBEM SDK Developer's Guide*.
- **XML services –** Convert XML data into CIM classes, enabling XML/HTTP-based WBEM clients to communicate with the CIM Object Manager.

Once connected to a WBEM-enabled system, WBEM clients can request WBEM operations such as creating, viewing, and deleting CIM classes and instances, querying for properties that have a specified value, and enumerating (getting a list of) instances or classes in a specified class hierarchy.

Software Components

Solaris WBEM Services software consists of three software components: Application, Management, and Provider. These components interact with the operating system and hardware. The following figure shows the software components and how they interact.

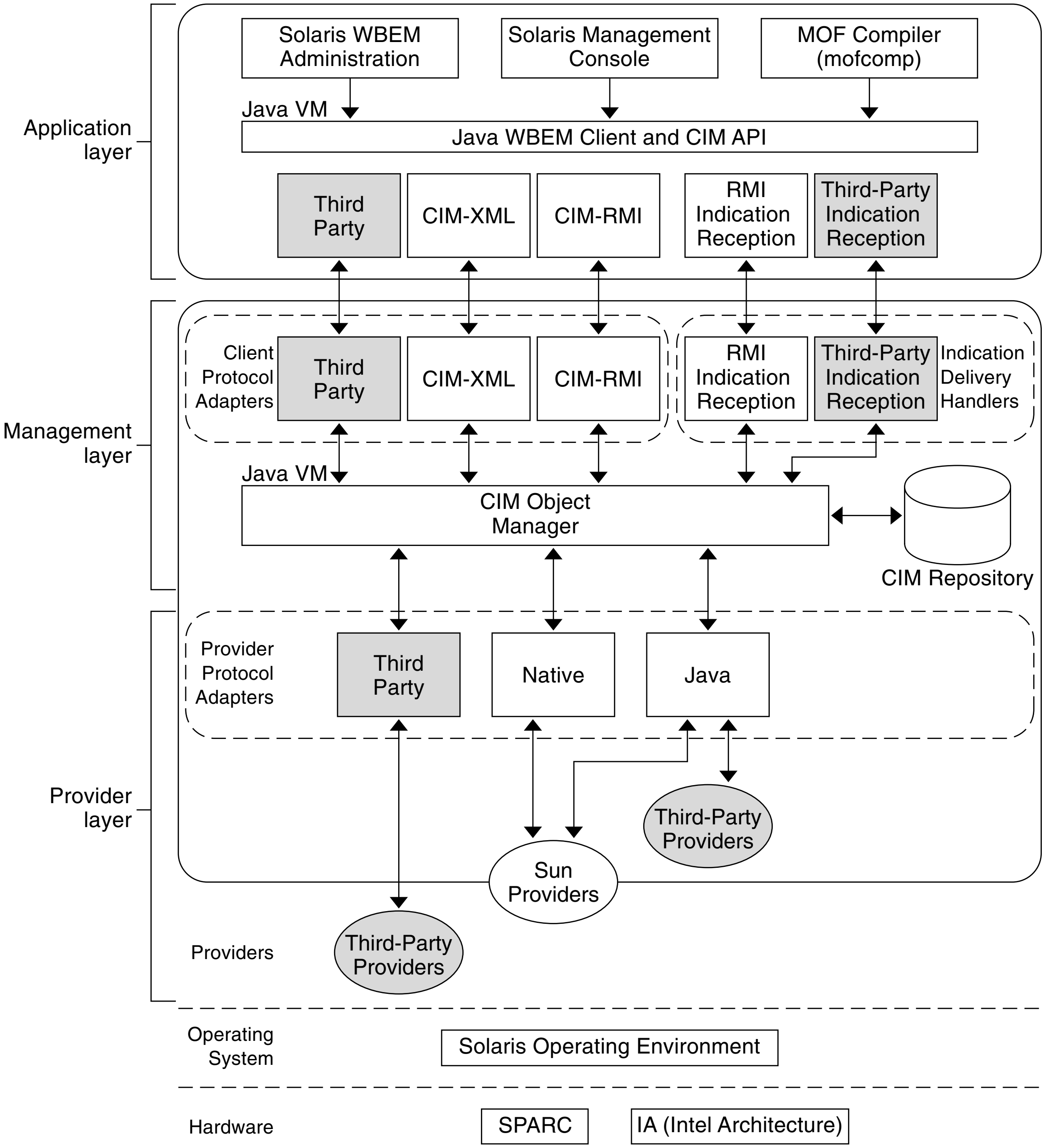


FIGURE 1–1 Solaris WBEM Services Architecture

- **Application layer –** WBEM clients process and display data from managed resources. Solaris WBEM Services includes the following applications.

    - **Sun WBEM User Manager and Solaris Management Console User tool –** Applications that allow system administrators to add and delete authorized users and to set these users' access privileges to managed resources.
    - **Solaris Management Console Log Viewer –** An application that displays log files. A user can view details of a log record, including the name of the user who issued a logged command and the client computer on which a logged event occurred.
    - **Managed Object Format (MOF) compiler –** A program that parses a file containing MOF statements, converts the classes and instances defined in the file to Java classes, and then adds the Java classes to the CIM Object Manager Repository, a central storage area for management data.

      MOF is a language for defining CIM classes and instances. MOF files are ASCII text files that use the MOF language to describe CIM objects. A CIM object is a representation, or model, of a managed resource, such as a printer, disk drive, or CPU. MOF files are located in `/usr/sadm/mof`.

      Many sites store information about managed resources in MOF files. Because MOF can be converted to Java, applications that can run on any system with a Java virtual machine can interpret and exchange this information. You can also use the `mofcomp` command to compile MOF files at any time after installation. MOF is described on the DMTF web page at `http://www.dmtf.org`.
- **Management layer –** Components at this layer provide services to connected WBEM clients.
    - **Common Information Model (CIM) Object Manager –** Software that manages CIM objects on a WBEM system. CIM objects are stored internally as Java classes. The CIM Object Manager transfers information between WBEM clients, the CIM Object Manager Repository, and managed resources.
    - **CIM Object Manager Repository –** Central storage area for CIM class and instance definitions.
    - **Client and CIM application programming interfaces (APIs) –** WBEM client applications use these Java interfaces to request operations, such as creating or viewing classes or instances of managed resources, from the CIM Object Manager.
    - **Provider interfaces –** Providers use these interfaces to transfer information about managed resources to the CIM Object Manager. The CIM Object Manager uses the provider interfaces to transfer information to locally installed providers.
- **Provider layer –** Providers act as intermediaries between the CIM Object Manager and one or more managed resources. When the CIM Object Manager receives a request from a WBEM client for data that is not available from the CIM Object Manager Repository, it forwards the request to the appropriate provider.
    - **Solaris providers –** Provide the CIM Object Manager with instances of managed resources in the Solaris operating environment. Providers get and set information on managed devices. A native provider is a machine-specific program written to run on a managed device. For example, a provider that

accesses data on a system running the Solaris operating environment probably includes C functions to query that system. The Java Native Interface is part of the JDK™ software. By writing programs using the Java Native Interface, you ensure that your code is portable across all platforms. The Java Native Interface enables Java code that runs within a Java virtual machine to operate with applications and libraries written in other languages, such as C, C++, and assembly.

  - **Solaris Schema –** A collection of classes that describes managed objects in the Solaris operating environment. The CIM Schema and Solaris Schema classes are stored in the CIM Object Manager Repository. The CIM Schema is a collection of class definitions used to represent managed objects that occur in every management environment.

    The Solaris Schema is a collection of class definitions that extend the CIM Schema and represent managed objects in a typical Solaris operating environment. Users can also use the MOF compiler (`mofcomp`) to add CIM Schema, Solaris Schema, or other classes to the CIM Object Manager Repository.

- **Operating system layer –** The Solaris providers enable management applications to access information about managed resources (devices and software) in the Solaris operating environment.
- **Hardware layer –** A management client can access management data on any supported Solaris platform.

Namespaces

One or more schemas can be stored in directory-like structures called namespaces. A CIM namespace is a directory-like structure that can contain other namespaces, classes, instances, and qualifier types. The names of objects within a namespace must be unique.

In Solaris WBEM Services, when a WBEM client application connects to a particular namespace, all subsequent operations occur within that namespace. When connected to a namespace, the client can access the classes and instances in that namespace (if they exist) and in any namespaces contained in that namespace. For example, if you create a namespace called `child` in the `root\cimv2` namespace, you could connect to `root\cimv2` and access the classes and instances in the `root\cimv2` and `root\cimv2\child` namespaces.

An application can connect to a namespace within a namespace. This is similar to changing to a subdirectory within a directory. Once the application connects to the new namespace, all subsequent operations occur within that namespace. If you open a new connection to `root\cimv2\child`, you can access any classes and instances in that namespace but cannot access the classes and instances in the parent namespace, `root\cimv2`.

The following namespaces are created by default during installation.

- `root` – The top-level namespace that contains other namespaces.
- `root\cimv2` – Contains the default CIM classes and instances that represent objects on your system, such as, `LogicalDisk` and `Netcard`. This is the default namespace.
- `root\security` – Contains the security classes used by the CIM Object Manager to represent access rights for users and namespaces.
- `root\snmp` – Contains the classes for the SNMP Provider and the SNMP Adapter for WBEM.
- `root\system` – Contains CIM Object Manager information and provider paths.

Providers

When a WBEM client application accesses CIM data, the WBEM system validates the user's login information on the current host. By default, a user is granted read access to the CIM Schema and the Solaris Schema. The CIM Schema describes managed objects on your system in a standard format that all WBEM-enabled systems and applications can interpret.

Providers are classes that communicate with managed objects to access data. Providers forward this information to the CIM Object Manager for integration and interpretation. When the CIM Object Manager receives a request from a management application for data that is not available from the CIM Object Manager Repository, it forwards the request to a provider.

The CIM Object Manager uses object provider APIs to communicate with providers. When an application requests dynamic data from the CIM Object Manager, the CIM Object Manager uses the provider interfaces to pass the request to the provider.

Providers perform the following functions in response to a request from the CIM Object Manager:

- Map the native information format to CIM classes
  - Get information from a device
  - Pass the information to the CIM Object Manager in the form of CIM classes
- Map the information from CIM classes to native device format
  - Get the required information from the CIM class
  - Pass the information to the device in native device format

Interoperability With Other WBEM Systems

A WBEM client and WBEM system can run on the same system or on different systems. Multiple WBEM clients can establish connections to the same WBEM system. For example, a WBEM system can serve four or five WBEM clients.

Solaris WBEM Services supports the Version 1.1 Specification for CIM Operations over HTTP. This specification uses XML to model CIM objects and messages. XML is a standard markup language for describing data on the Web. This standard extends XML markup to define CIM objects and operations. Because XML provides a standard way of describing data that can be sent across the Web, any WBEM client can access CIM data on any WBEM system that can parse XML data.

Solaris WBEM Software Developer's Kit

The Solaris WBEM Software Developer's Kit (SDK) contains the components required to write management applications that can communicate with any WBEM-enabled management device. Developers can also use this SDK to write providers, which are programs that communicate with managed objects to access data. All management applications developed using the Solaris WBEM SDK run in the Java environment.

A WBEM client application is a program that uses Solaris WBEM APIs to manipulate CIM objects. A client application typically uses the CIM API to construct an object (for example, a namespace, class, or instance) and then to initialize that object. The application then uses the client APIs to pass that object to the CIM Object Manager and to request a WBEM operation, such as operations that create a CIM namespace, class, or instance.

The Solaris WBEM SDK installs and runs in the Java environment. You can use the Solaris WBEM SDK as a standalone application or with Solaris WBEM Services.

The Solaris WBEM SDK is described in the *Solaris WBEM SDK Developer's Guide*.

CHAPTER **2**

Using the CIM Object Manager (Tasks)

The Common Information Model (CIM) Object Manager is software that transfers CIM data between WBEM client applications and managed resources.

Here is a list of the information in this chapter.

- "Upgrading the CIM Object Manager Repository (Task Map)" on page 28
- "About the CIM Object Manager" on page 25
- "`init.wbem` Command" on page 26
- "Stopping and Restarting the CIM Object Manager" on page 27
- "Exception Messages" on page 32

About the CIM Object Manager

The CIM Object Manager manages CIM objects on a WBEM-enabled system. A CIM object is a representation, or model, of a managed resource, such as a printer, disk drive, or CPU. CIM objects are stored internally as Java classes.

When a WBEM client application accesses information about a CIM object, the CIM Object Manager contacts either the appropriate provider for that object or the CIM Object Manager Repository. Providers are classes that communicate with managed objects to access data. When a WBEM client application requests data from a managed resource that is not available from the CIM Object Manager Repository, the CIM Object Manager forwards the request to the provider for that managed resource. The provider dynamically retrieves the information.

At startup, the CIM Object Manager performs the following functions:

- Listens for RMI connections on RMI port 5987 and for XML/HTTP connections on HTTP port 5988
- Sets up a connection to the CIM Object Manager Repository

- Waits for incoming requests

The CIM Object Manager:

- Performs security checks to authenticate user login and authorization to access namespaces
- Performs syntactical and semantic checking of CIM data operations to ensure that they comply with the latest CIM Specification
- Routes requests to the appropriate provider or to the CIM Object Manager Repository
- Delivers data from providers and from the CIM Object Manager Repository to WBEM client applications

A WBEM client application contacts the CIM Object Manager to establish a connection when it needs to perform WBEM operations, such as creating a CIM class or updating a CIM instance. When a WBEM client application connects to the CIM Object Manager, the WBEM client gets a reference to the CIM Object Manager, which it then uses to request services and operations.

`init.wbem` Command

Solaris automatically runs `init.wbem` during installation and every time you reboot a system. The `init.wbem` command starts the CIM Object Manager and Solaris Management Console server, both of which run combined in a single process. You can also use `init.wbem` to stop the CIM Object Manager, to stop the Solaris Management Console server, or to retrieve status from a server. You can find additional information about this command in the `init.wbem`(1M) man page.

Generally, you do not need to stop the CIM Object Manager. However, if you change an existing provider, you must stop and restart the CIM Object Manager before using the updated provider.

You can specify three options with `init.wbem`:

- `start` – Starts the CIM Object Manager or Solaris Management Console server on the local host.
- `stop` – Stops the CIM Object Manager and Solaris Management Console server on the local host.
- `status` – Gets status for the CIM Object Manager and Solaris Management Console server on the local host.

Solaris Management Console Server

The Solaris Management Console software provides Solaris management applications such as User Manager, Disk Manager, and Log Viewer. The Solaris Management Console server provides tools for the console to download and performs common services for the console and its tools, such as authentication, authorization, logging, messaging, and persistence.

The Solaris Management Console is described in other chapters in this document, and is also described in *System Administration Guide: Basic Administration*.

System Booting

The `init.wbem` command is located in the `/etc/init.d` directory. The file `/etc/rc2.d/S90wbem` runs with the `start` option when initialization state 2 is entered, normally at boot time. The files `/etc/rc0.d/K36wbem`, `/etc/rc1.d/K36wbem`, and `/etc/rcS.d/K36wbem` are run with the `stop` option when initialization states 0, 1, and S are entered, normally when the system halts, or when the system enters either system administrator mode or single-user mode.

Stopping and Restarting the CIM Object Manager

If you change a provider, you must stop and restart the CIM Object Manager before using the updated provider.

▼ How to Stop the CIM Object Manager

1. **Become superuser.**
2. **Stop the CIM Object Manager.**

```
# /etc/init.d/init.wbem stop
```

▼ How to Restart the CIM Object Manager

1. **Become superuser.**
2. **Restart the CIM Object Manager.**

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

Upgrading the CIM Object Manager Repository (Task Map)

The following table identifies the procedures to upgrade the CIM Object Manager Repository. Whether you save the JavaSpaces datastore and convert or merge WBEM data depends on the version of the Solaris operating environment that you are using before you upgrade to the Solaris 9 operating environment. Table 2–1 describes how to determine the procedures you follow to upgrade the CIM Object Manager Repository.

Task	Description	For Instructions
Save the JavaSpaces datastore.	Save the JavaSpaces datastore by downloading or copying files and determining the version of the JDK that is currently installed on your system.	"How to Save the JavaSpaces Datastore" on page 29
Convert WBEM data.	Convert WBEM data by using the `wbemconfig convert` command.	"How to Convert WBEM Data" on page 30
Merge WBEM data.	Merge WBEM data by using the `wbemconfig convert` command.	"How to Merge WBEM Data" on page 31

Upgrading the CIM Object Manager Repository

You must update any proprietary custom Managed Object Format (MOF) data to the new Reliable Log repository format that is used with WBEM Services 2.5 in Solaris 9.

Before you upgrade to the Solaris 9 operating environment, you might need to save the JavaSpaces™ datastore. After you upgrade, you must convert or merge data, depending on the version of the Solaris environment that you were running before you upgraded to the Solaris 9 environment.

Failure to convert or merge the data results in data loss.

Use the following table to determine whether or not to save the JavaSpaces software before you upgrade and whether to convert or merge the WBEM data after you upgrade to the Solaris 9 operating environment.

TABLE 2–1 Determining Whether to Convert or Merge WBEM Data

Operating Environment Before Upgrading to Solaris 9	Save JavaSpaces Datastore Before You Upgrade?	Convert or merge?
Solaris 8 (Solaris WBEM Services 2.0)	Yes	Convert
Solaris 8 6/00 (WBEM Services 2.0)		
Solaris 8 10/00 (WBEM Services 2.2)		
Solaris 8 1/01 (WBEM Services 2.3)	No	Merge
Solaris 8 4/01 (WBEM Services 2.4)		
Solaris 8 7/01 (WBEM Services 2.4)		
Solaris 8 10/01 (WBEM Services 2.4)		
Solaris 9 (Beta) (WBEM Services 2.5)		

▼ How to Save the JavaSpaces Datastore

1. **Become superuser.**

2. **Do you want to download the files that you will need, or do you want to save your current JavaSpaces datastore?**

Note – The safer method is to save your JavaSpaces datastore rather than to download files.

- If you want to download the files, go to “How to Convert WBEM Data” on page 30.
- If you want to save your JavaSpaces datastore, enter the following commands:

```
# cd /usr/sadm/lib/wbem
# cp outrigger.jar outrigger.jar.tmp
# cp outrigger-dl.jar outrigger-dl.jar.tmp
# cp transient-outrigger.jar transient-outrigger.jar.tmp
# cp jini-core.jar jini-core.jar.tmp
# cp jini-ext.jar jini-ext.jar.tmp
# cp tools.jar tools.jar.tmp
# cp pro.zip pro.zip.tmp
```

3. **Determine and record the version of the JDK that is currently installed on your system.**

```
# /usr/bin/java -version
java version "1.2.1"
Solaris VM (build Solaris_JDK_1.2.1_04c, native threads, sunwjit)
```

Note – You must be running the same version of the JDK as you used when you created the original JavaSpaces datastore to convert WBEM data.

▼ How to Convert WBEM Data

1. **Upgrade your system to the Solaris 9 operating environment.**

2. **Become superuser.**

3. **Stop the CIM Object Manager.**

   ```
   # /etc/init.d/init.wbem stop
   ```

Caution – Failure to stop the CIM Object Manager before running `wbemconfig convert` might corrupt your data.

---

4. **Did you save your current JavaSpaces datastore in "How to Save the JavaSpaces Datastore" on page 29?**

   - If yes, restore your JavaSpaces datastore.

     ```
     # cd /usr/sadm/lib/wbem
     # cp outrigger.jar.tmp outrigger.jar
     # cp outrigger-dl.jar.tmp outrigger-dl.jar
     # cp transient-outrigger.jar.tmp transient-outrigger.jar
     # cp jini-core.jar.tmp jini-core.jar
     # cp jini-ext.jar.tmp jini-ext.jar
     # cp tools.jar.tmp tools.jar
     # cp pro.zip.tmp pro.zip
     ```

   - If no, download and unzip the file `UpgradeRepository.zip` from `http://www.sun.com/solaris/wbem`.

     `UpgradeRepository.zip` contains the `.jar` files that you need to later convert the WBEM data.

5. **In a directory other than the one in which the JDK you are currently using is installed, obtain and install the JDK that you recorded in "How to Save the JavaSpaces Datastore" on page 29.**

6. **Change the symbolic link from the currently installed JDK in `/usr/java` to the JDK you recorded in "How to Save the JavaSpaces Datastore" on page 29.**

   For example, to change the currently installed JDK to `Solaris_JDK_1.2.1_04c` in `/old_sdk`, type:

```
# rm /usr/java
# ln -s /old_sdk/Solaris_JDK_1.2.1_04c /usr/java
```

7. **Convert the data in the JavaSpaces datastore to Reliable Log format.**

```
# /usr/sadm/lib/wbem/wbemconfig convert
```

The `wbemconfig convert` command successfully converts any proprietary custom MOF data, but not any CIM or Solaris MOF data that you have modified. CIM and Solaris MOF data that you have modified is destroyed.

Note – To recompile any modified CIM or Solaris MOF data in the new repository, use the `mofcomp` command to compile the MOF files that contain the class definitions.

---

8. **Change the symbolic link from `/usr/java` to the location of the JDK software that ships with the Solaris 9 operating environment.**

For example, to change the symbolic link from `/usr/java1.4`, type:

```
# rm /usr/java
# ln -s /usr/java1.4 /usr/java
```

9. **Stop the CIM Object Manager.**

```
# /etc/init.d/init.wbem stop
```

10. **Start the CIM Object Manager.**

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

The CIM Object Manager adds repository files that contain the converted data to the directory `/var/sadm/wbem/logr/`, which the Solaris installer created when you upgraded your system to Solaris 9.

▼ How to Merge WBEM Data

1. **Upgrade your system to the Solaris 9 operating environment.**

2. **Become superuser.**

3. **Stop the CIM Object Manager.**

```
# /etc/init.d/init.wbem stop
```

Caution – Failure to stop the CIM Object Manager before you run `wbemconfig convert` might corrupt your data.

---

4. **Merge the original data in the previous Reliable Log with the data in the Solaris 9 Reliable Log.**

```
# /usr/sadm/lib/wbem/wbemconfig convert
```

Note – The `wbemconfig convert` command successfully converts any proprietary custom MOF data, but not any CIM or Solaris MOF data that you have modified. CIM and Solaris MOF data that you have modified is destroyed. To recompile any modified CIM or Solaris MOF data in the new repository, use the `mofcomp` command to compile the MOF files that contain the class definitions.

Exception Messages

The CIM Object Manager generates exception messages to indicate incorrect MOF syntax and semantics. The *Solaris WBEM SDK Developer's Guide* contains information about exception messages.

CHAPTER **3**

Using SNMP Services for WBEM (Tasks)

SNMP Services for WBEM includes two components, the SNMP Adapter for WBEM and the SNMP Provider.

Here is a list of the information in this chapter.

- "Installing and Using the SNMP Adapter for WBEM (Task Map)" on page 43
- "SNMP Adapter for WBEM" on page 33
- "How the SNMP Adapter for WBEM Works" on page 34
- "Configuring the Adapter and Mapping SNMP to CIM Objects" on page 35
- "Troubleshooting Problems With the SNMP Adapter for WBEM" on page 45
- "SNMP Provider" on page 49

SNMP Adapter for WBEM

Intended for use by system administrators, the SNMP Adapter for WBEM enables Simple Network Management Protocol (SNMP) management applications to access system management information that is provided by Solaris WBEM Services.

Used with `snmpdx`, the Master Agent of the Solstice Enterprise Agents™ technology, the SNMP Adapter for WBEM maps SNMP requests into equivalent WBEM Common Information Model (CIM) properties or instances. The Master Agent is described in `snmpdx`(1M).

The SNMP Adapter for WBEM also remaps the response from the CIM Object Manager into an SNMP response, which is returned to the management application.

A mapping file contains the corresponding Object Identifier (OID), class name, property name, and Abstract Syntax Notation One (ASN.1) type for each object. You can create your own mapping files.

How the SNMP Adapter for WBEM Works

The Solaris operating environment initializes WBEM Services before starting the Solstice Enterprise Agents Master Agent. By default, the SNMP Adapter for WBEM, or Adapter, is disabled. However, once you enable the Adapter, the Solstice Enterprise Agents Master Agent (snmpdx) starts the Adapter automatically. The Adapter is described in snmpXwbemd(1M).

An SNMP Manager passes an SNMP Get-request to the Solstice Enterprise Agents Master Agent. The Master Agent then sends the Get-request to the Adapter, which uses the mapping files in /var/sadm/wbem/snmp/map to translate the objects in the Get-request into corresponding CIM objects. The Adapter also translates the CIM objects into SNMP objects in a Get-response.

Note – At present, only Get-request and scalar objects are supported in Solaris 9. Get-next-request, Get-bulk-request, and Set-request as well as other objects are not currently supported.

The Adapter searches this directory alphabetically for the first file to which the extension .map is appended. The Adapter then reads all mapping files in the directory and caches their contents. The Adapter uses the contents of these files to translate the objects that are specified in the Get-request into corresponding CIM objects. The Adapter subsequently ignores duplicate OIDs in the mapping files in the directory. For example, if this OID appears in 002SUNWlvma.map:

```
1.3.6.1.2.1.1.1.0 My_ComputerSystem Description SnmpString
```

and the same OID appears in 050SUNWwbcou.map, which the Adapter reads after 002SUNWlvma.map:

```
1.3.6.1.2.1.1.1.0 Solaris_ComputerSystem Description SnmpString
```

then the Adapter ignores the OID that is specified in 050SUNWwbcou.map.

The Adapter subsequently generates a Get-response for each Get-request that an SNMP Manager submits. If the Adapter cannot find a corresponding entry in any mapping file, the Adapter returns a Get-response error.

How the Master Agent Routes a Request: SNMP Adapter for WBEM Compared to the Sun SNMP Agent

Until the release of the SNMP Adapter for WBEM, when an SNMP Manager sent a Get-request for an SNMP MIB-2 variable to the Solstice Enterprise Agents Master Agent, the Master Agent routed the request to the Sun SNMP MIB-2 Agent (`mibiisa`). Because the Adapter also handles SNMP MIB-2 requests, what happens if the Sun SNMP Agent and the SNMP Adapter for WBEM are both running at the same time? How does the Master Agent route a request?

The Master Agent builds a node table based on the subtrees that are defined in each subagent registration file. The `mibiisa` subagent registers the entire MIB-2 subtree and the Sun Microsystems MIB subtree. The Adapter registers the `MIB-2.system` subtree and the `hostRsrc` subtree. The Master Agent does not allow two agents to register the same subtree.

The Sun SNMP MIB-2 Agent is described in `mibiisa`(1M). The Master Agent is described in the *Solstice Enterprise Agents 1.0 User Guide*.

At initialization, the Master Agent creates a node table that contains each subtree that is registered. The Master Agent forwards each Get-request to the agent whose subtree best matches the OID that is included in the request. A request for `mib-2.system.5.0`, for example, is forwarded to the Adapter. A request for `mib-2.interfaces.1.0`, on the other hand, is forwarded to the `mibiisa` subagent. If the OID is not defined within any subtree that is registered by the Master Agent, the Master Agent returns an error in the Get-response.

The SNMP Adapter for WBEM supports SNMP V1 requests only.

Configuring the Adapter and Mapping SNMP to CIM Objects

Configuration Files

The files you use to configure the SNMP Adapter for WBEM, which are located in `/etc/snmp/conf/`, are described in this section.

In `snmpXwbem.acl`, you define the access control list policies that are associated with the Adapter, in this format:

```
#pragma ident  "@(#)snmpXwbem.acl   1.2   01/04/18 SMI"
#Copyright (c) 2001 by Sun Microsystems, Inc.
#All rights reserved.

#        Configuration file of the SNMP subagent for WBEM

##################
# access control #
##################
# The list of community names needed for read/write access
# to the entire MIB.

# If the list is empty, the only valid community name is "public"
# and its access type is read-only
#
# A * in the managers list indicates requests can be received from
# any host.

acl = {
        {
                communities = public, private
                access = read-only
                managers = *
        }
}

###################
# trap parameters #
###################
trap = {
}
```

A comma-separated list of communities and a comma-separated list of managers are allowed. The access policies are read-only. An empty `trap` clause is required. Traps are not supported by the Adapter in Solaris 9.

In `snmpXwbem.reg`, you define the Object Identifier (OID) of the subtree for which the Adapter is responsible, in this format:

```
#pragma ident   "@(#)snmpXwbem.reg      1.3    01/10/04 SMI"
#
#Copyright (c) 2001 by Sun Microsystems, Inc.
#All rights reserved.

#       Configuration file of the SNMP subagent for WBEM

##########
# macros #
##########

# The following 3 macros are predefined:
#
#       mib-2 =         1.3.6.1.2.1
#       enterprise =    1.3.6.1.4.1
#       sun =           1.3.6.1.4.1.42
#
# You can define your own macros, so that you can
# manipulate strings instead of OIDs in defining the agent.
# See the "agent" section below.

macros = {
        system = mib-2.1
        hostRsrc = mib-2.25
}


##########
# agent  #
##########

# You must fill in at least the following fields:
#
# - name:               the name of your agent (for example, the executable
#                       file name of your agent)
#
# - subtrees:           the list of OIDs / subtrees of OIDs your agent
#                       supports. The listed items must be separated by
#                       a comma.
#
# You can also change or add the following fields:
#
# - timeout:            the number of micro-seconds the SNMP Relay will
#                       wait for a response from your agent
#
# - watch-dog-time:     the number of seconds the SNMP Relay will wait to
#                       test whether the subagent is active, if there has
#                       been no activity for the watch-dog-time interval
#
# - port:               the UDP port number on which you will start
#                       your agent

agents =
{
        {
                name = "WBEMsubagent"
                subtrees = { system, hostRsrc }
                timeout = 20000000
                watch-dog-time = 240
        }
}
```

The unit of measure for timeout is microseconds. The unit of measure for watch-dog-time is seconds. By default, the Master Agent tries to start the Adapter every four minutes (or number of seconds to which watch-dog-time is set).

Note – The Master Agent automatically determines the port to be used by the Adapter.

In snmpXwbem.rsrc-, you define a pointer to the registration file and you define how the SNMP Master Agent is to start the Adapter, in this format:

```
#pragma ident   "@(#)snmpXwbem.rsrc-   1.2   01/04/18 SMI"
#Copyright (c) 2001 by Sun Microsystems, Inc.
#All rights reserved.

#        Configuration file of the SNMP subagent for WBEM

##########
# agents #
##########
resource =
{
        {
                registration_file = "/etc/snmp/conf/snmpXwbem.reg"
                security = "/etc/snmp/conf/snmpXwbem.acl"
                policy = "spawn"
                type = "legacy"
                command = "/usr/sadm/lib/wbem/snmpXwbemd -p $PORT"
        }
}
```

Mapping Files

When the Master Agent passes a Get-request to the SNMP Adapter for WBEM, the Adapter uses the mapping files in /var/sadm/wbem/snmp/map to translate the Get-request into a CIM object request. The Solaris operating environment includes a mapping file for you in this directory. You can also define your own mapping file for the CIM instrumentation that you want to view through an SNMP Manager.

This section shows the contents of a mapping file that the Solaris environment provides, and describes what you need to know to create an Adapter mapping file.

Contents of the Mapping File That Is Included in Solaris

This example shows the contents of the mapping file that the Solaris operating environment includes for you:

```
#
#pragma ident    "@(#)050SUNWwbcou.map    1.0    01/04/03 SMI"
#
# Copyright (c) 2001 by Sun Microsystems, Inc.
# All rights reserved.
```

```
#
# *** Description of contents ***
#
# First non-commented non-blank line contains required Version label.
# Remaining non-commented non-blank lines are considered map entries
# used as described below:
#
# Column 1 - SNMP OID - Uniquely describes an SNMP variable
# Column 2 - CIM Class Name - CIM class associated with this variable
# Column 3 - CIM Property Name - CIM property that maps to SNMP OID variable
# Column 4 - ASN.1 type - SNMP datatype that dictates how data is mapped
#            to/from SNMP requests.  Supported types are: SnmpString, SnmpOid,
#            SnmpTimeticks, SnmpCounter, SnmpInt, SnmpGauge, SnmpIpAddress,
#            SnmpOpaque)
# Column 5 and greater are ignored
#
Version 1.0

1.3.6.1.2.1.1.1.0 Solaris_ComputerSystem Description SnmpString
1.3.6.1.2.1.1.3.0 Solaris_OperatingSystem LastBootUpTime SnmpTimeticks
1.3.6.1.2.1.1.4.0 Solaris_ComputerSystem PrimaryOwnerContact SnmpString
1.3.6.1.2.1.1.5.0 Solaris_ComputerSystem Name SnmpString

1.3.6.1.2.1.25.1.5.0 Solaris_OperatingSystem NumberOfUsers SnmpGauge
1.3.6.1.2.1.25.1.6.0 Solaris_OperatingSystem NumberOfProcesses SnmpGauge
1.3.6.1.2.1.25.1.7.0 Solaris_OperatingSystem MaxNumberOfProcesses SnmpGauge
1.3.6.1.2.1.25.1.2.0 Solaris_OperatingSystem LocalDateTime SnmpString
```

The contents of this mapping file associate the SNMP MIB-2 System Group scalar objects with their corresponding CIM objects.

MIB-2 System Group Scalar Object	CIM Object
sysDescr	Solaris_ComputerSystem.Description
sysUpTime	Solaris_OperatingSystem.LastBootUpTime
sysContact	Solaris_ComputerSystem.PrimaryOwnerContact
sysName	Solaris_ComputerSystem.Name

The contents of this mapping file also associate the SNMP Host Resources MIB objects with their corresponding CIM objects.

SNMP Host Resources MIB Object	CIM Object
hrSystemNumUsers	Solaris_OperatingSystem.NumberOfUsers
hrSystemProcesses	Solaris_OperatingSystem.NumberOfProcesses

SNMP Host Resources MIB Object	CIM Object
`hrSystemMaxProcesses`	`Solaris_OperatingSystem.MaxNumberOfProcesses`
`hrSystemDate`	`Solaris_OperatingSystem.LocalDateTime`

The syntax of the contents of a mapping file is described in "Syntax of the Contents of a Mapping File" on page 40.

Note – At present, the only way to retrieve host resource data is through the CIM Object Manager, as Solaris does not currently provide an SNMP Host Resource agent.

Syntax of a Mapping File Name

To ensure that the Adapter reads your mapping file, name the file according to this syntax:

alphanumeric-string`.map`

alphanumeric-string represents an alphanumeric string. For example, here is the name of the mapping file that Solaris includes:

`050SUNWwbcou.map`

You include the three digits to ensure that the Adapter reads the files in a more precise order. For example, `002SUNWlvma.map` is read before `050SUNWwbcou.map`.

Note – You must allow `root` to at least read the mapping files that you create.

```
$ chmod 400 002SUNWlvma.map
```

Syntax of the Contents of a Mapping File

The following table describes the elements and the syntax of the contents of a mapping file.

TABLE 3–1 Contents of a Mapping File

Element	Description	Required?
`#`	A comment, which is always ignored.	No
`Version 1.0`	The version of the mapping file. The text string that specifies the version must be the first uncommented line. If you do not specify a version as shown, the mapping file is ignored.	Yes
`1.3.6.1.2.1.1.1.0`	The SNMP Object Identifier, or OID, which is the key you want to extract from the SNMP request. The SNMP OID describes an SNMP variable. Because the Adapter currently supports scalars only, the OID must end with the text string `.0`.	Yes
`Solaris_ComputerSystem`	The CIM class name that is associated with the variable.	Yes
`Description`	The CIM property name that defines a characteristic of the specified class and that maps to the SNMP OID variable.	Yes

TABLE 3–1 Contents of a Mapping File *(Continued)*

Element	Description	Required?
SnmpString	The ASN.1 data type. Values that you can specify, including how they are mapped, are: ■ SnmpString – Move string, number, or CIM LocalDateTime into SnmpString. ■ SnmpInt – Move CIM number data types (including a string in number format) into SnmpInt (signed, 32-bit integer). ■ SnmpCounter – Move CIM number data types (including string in number format) into SnmpCounter (unsigned, 32-bit integer). ■ SnmpGauge – Move CIM number data types (including string in number format) into SnmpGauge (unsigned, 32-bit integer). ■ SnmpTimeticks – Move the time difference, represented in hundredths of a second, into SnmpTimeticks. This value is calculated by subtracting the CIM value from the current time. For example, sysUpTime is calculated by subtracting bootTime from currentTime. ■ SnmpIpAddress – Move string into SnmpIpAddress. You must specify the string in IP address format. ■ SnmpOid – Move string into SnmpOid. You must specify the string in OID format. ■ SnmpOpaque – Move vector of bytes into SnmpOpaque.	Yes

Installing and Using the SNMP Adapter for WBEM (Task Map)

The following table identifies the procedures that you need to follow to install, start, stop, and use the SNMP Adapter for WBEM, the Adapter.

Task	Description	For Instructions
Install the SNMP Adapter for WBEM.	Install the Adapter when you install the Solaris operating environment.	"How to Install the SNMP Adapter for WBEM" on page 43
Start the SNMP Adapter for WBEM.	Start the SNMP Adapter for WBEM by moving `snmpXwbem.rsrc-` into `snmpXwbem.rsrc`.	"How to Start the SNMP Adapter for WBEM" on page 44
Disable or stop the SNMP Adapter for WBEM.	Stop the SNMP Adapter for WBEM by using the `pkill` command.	"How to Disable the SNMP Adapter for WBEM" on page 44
Force the SNMP Adapter for WBEM to reread the mapping file directory.	Force the SNMP Adapter for WBEM to reread the mapping file directory by updating `snmpXwbem.reg` so that it includes the new subtree, and by using the `pkill` command.	"How to Force the SNMP Adapter for WBEM to Reread the Mapping File Directory" on page 44

Installing and Using the SNMP Adapter for WBEM

This section describes how to install, start, stop, and use the SNMP Adapter for WBEM.

▼ How to Install the SNMP Adapter for WBEM

- **Install the Solaris operating environment on your system.**

  The Adapter software is installed on your system along with the Solaris software.

▼ How to Start the SNMP Adapter for WBEM

When you are ready to retrieve data from the CIM Object Manager through your SNMP application, follow these steps to start the Adapter.

1. **Become superuser.**

2. **Stop the Master Agent.**

   ```
   # /etc/init.d/init.snmpdx stop
   ```

3. **Change directory to** `/etc/snmp/conf`**.**

4. **Move** `snmpXwbem.rsrc-` **into the Adapter resource file.**

   ```
   # mv snmpXwbem.rsrc- snmpXwbem.rsrc
   ```

5. **Restart the Master Agent.**

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

▼ How to Disable the SNMP Adapter for WBEM

When you are finished retrieving data from the CIM Object Manager or to modify a file in `/etc/snmp/conf`, you need to disable the Adapter. Follow these steps.

1. **Become superuser.**

2. **Stop the Master Agent.**

   ```
   # /etc/init.d/init.snmpdx stop
   ```

3. **Stop the Adapter.**

   ```
   # /usr/bin/pkill -9 -x -u 0 snmpXwbemd
   ```

4. **Change directory to** `/etc/snmp/conf`**.**

5. **Temporarily rename** `snmpXwbem.rsrc`**.**

   ```
   # mv snmpXwbem.rsrc snmpXwbem.rsrc-
   ```

6. **Restart the Master Agent.**

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

▼ How to Force the SNMP Adapter for WBEM to Reread the Mapping File Directory

After you place a new mapping file or update an existing mapping file in `/var/sadm/wbem/snmp/map`, you must signal the Adapter to reread all mapping files in the directory. You specify the signal SIGHUP to signal the Adapter.

Follow these steps to force the Adapter to reread all mapping files (without having to restart the CIM Object Manager).

1. **Become superuser.**
2. **Does a new mapping file or a new entry in a mapping file reference a subtree that is not registered by the Adapter?**
   - If yes, go to the next step.
   - If no, go to Step 5.
3. **Update** `/etc/snmp/conf/snmpXwbem.reg` **so that it includes the new subtree.**
4. **Stop and restart the Master Agent.**

```
# /etc/init.d/init.snmpdx stop
# /etc/init.d/init.snmpdx start
```

5. **Signal the Adapter that you have updated a mapping file.**

```
# /usr/bin/pkill -1 -x -u 0 snmpXwbemd
```

Troubleshooting Problems With the SNMP Adapter for WBEM

This section lists specific console error messages that you might encounter when using the Adapter.

If you encounter errors, problems, or unexpected results that are not described in this section or if you want to troubleshoot problems more precisely, use the Solaris Management Console Log Viewer to view log data.

Instructions that describe how to start the Solaris Management Console Log Viewer are presented in "Viewing Log Data Through Log Viewer" on page 65.

Sending and Receiving Requests

Error Message

```
ERROR: sending request to Adapter Service.
```

```
ERROR: receiving request from Adapter Service.
```

Cause

Either `snmpXwbemd` believes WBEM is enabled but cannot communicate with the Adapter Service, or the request timed out.

▼ Solution

Send another request. If sending another request fails, verify that the request and response FIFOs do not contain pending messages, that is, contain 0 bytes.

1. **Type:**

   ```
   # cd /var/sadm/wbem/snmp
   ```

2. **Type:**

   ```
   # ls -l
   ```

   The request and response FIFOs are listed.

3. **Does either FIFO contain pending messages (contain more than 0 bytes)?**

   If yes, go to step Step 4.

   If no:

   a. **Ensure that you need to stop WBEM by using the Solaris Management Console Log Viewer to view log data and determine the problem.**

      Instructions that describe how to start the Solaris Management Console Log Viewer are presented in "Viewing Log Data Through Log Viewer" on page 65.

   b. **If necessary, stop WBEM.**

      ```
      # /etc/init.d/init.wbem stop
      ```

   c. **Go to Step 9.**

4. **Stop WBEM.**

   ```
   # /etc/init.d/init.wbem stop
   ```

5. **Stop the Master Agent.**

   ```
   # /etc/init.d/init.snmpdx stop
   ```

6. **Change to the directory in which the FIFOs are located.**

   ```
   # cd /var/sadm/wbem/snmp
   ```

7. **Remove both FIFOs.**

   ```
   # rm _adapter_rcv.fifo
   # rm _adapter_snd.fifo
   ```

8. **Restart the Master Agent.**

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

9. **If you stopped WBEM in Step 3 or in Step 4, restart it.**

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

FIFO Cannot Be Opened

Error Message

```
ERROR: request FIFO cannot be opened.

ERROR: response FIFO cannot be opened.
```

Cause

A protocol problem occurred either when the Adapter received a request or when the Adapter processed a response.

▼ Solution

Send another request. If sending another request fails, verify that the request and response FIFOs do not contain pending messages, that is, contain 0 bytes.

1. **Type:**

   ```
   # cd /var/sadm/wbem/snmp
   ```

2. **Type:**

   ```
   # ls -l
   ```

   The request and response FIFOs are listed.

3. **Does either FIFO contain pending messages (contain more than 0 bytes)?**

   If yes, go to step Step 4.

   If no:

   a. **Ensure that you need to stop WBEM by using the Solaris Management Console Log Viewer to view log data and determine the problem.**

      Instructions that describe how to start the Solaris Management Console Log Viewer are presented in "Viewing Log Data Through Log Viewer" on page 65.

   b. **If necessary, stop WBEM.**

      ```
      # /etc/init.d/init.wbem stop
      ```

   c. **Go to Step 9.**

4. **Stop WBEM.**

   ```
   # /etc/init.d/init.wbem stop
   ```

5. **Stop the Master Agent.**

   ```
   # /etc/init.d/init.snmpdx stop
   ```

6. **Change to the directory in which the FIFOs are located.**

   ```
   # cd /var/sadm/wbem/snmp
   ```

7. **Remove both FIFOs.**

   ```
   # rm _adapter_rcv.fifo
   # rm _adapter_snd.fifo
   ```

8. **Restart the Master Agent.**

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

9. **If you stopped WBEM in Step 3 or in Step 4, restart it.**

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

FIFO Cannot Be Created

Error Message

```
ERROR: FIFO cannot be created.
```

Cause

A system error occurred when the Adapter attempted to create the request or the response FIFO.

Solution

Verify that /var/sadm/wbem/snmp exists and has write access.

WBEM Services Are Not Started

Error Message

```
ERROR: WBEM Services are not started.
```

Cause

The Master Agent cannot detect if WBEM Services have started and are running.

Solution

Restart WBEM and wait the number of seconds to which `watch-dog-time` in `snmpXwbem.reg` is set.

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

After a period of time, the Master Agent starts the Adapter automatically. By default, the Master Agent tries to start the Adapter every four minutes (or number of seconds to which `watch-dog-time` is set).

Note – If you do not want to wait for the Master Agent to start the Adapter automatically, stop and then restart the Master Agent.

```
# /etc/init.d/init.snmpdx stop
```

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

The Master Agent immediately starts the Adapter.

SNMP Provider

The SNMP Provider is a software component that provides information about managed elements to the CIM Object Manager, including configuration information about an SNMP device.

Note – The SNMP Provider supports traps by listening to port 162 for SNMP V1 and SNMP V2 traps. To display the events that are generated by these traps, you must subscribe to `CIM_SNMPTrapIndication`.

Generating a MOF File From an SNMP MIB File

When you want to access Simple Network Management Protocol (SNMP) information through the SNMP Provider, you use a Management Information Base (MIB) file to generate a Managed Object Format (MOF) file. The `mib2mof` command generates qualifiers that enable the SNMP Provider to map CIM operations that are performed on the CIM classes in the MOF file to SNMP operations. The `mib2mof` command is described in `mib2mof`(1M).

Note – The SNMP Provider supports SNMP traps. Traps are reported in the CIM process indication event `CIM_SNMPTrapIndication`. When a client subscribes to the provider for this event, the provider listens on port 162 for SNMP V1 and SNMP V2 traps. The information is copied from the trap to the indication. Then, the indication is delivered to the client.

The MOF files that describe the CIM Schema and the Solaris Schema are located in `/usr/sadm/mof`. MOF files are described in more detail in the *Solaris WBEM SDK Developer's Guide*.

▼ How to Generate a MOF File From an SNMP MIB File

1. **Become superuser.**
2. **Type the command:**

```
# mib2mof SNMP_MIB_filename
```

Example:

```
# /usr/sadm/bin/mib2mof sysctl.mib
```

CHAPTER **4**

Administering Security (Tasks)

This chapter describes WBEM security mechanisms and the features that the CIM Object Manager enforces.

Here is a list of the information in this chapter.

WBEM Security Mechanisms

WBEM employs several mechanisms to ensure secure access to its data, including:

- **Authentication –** The process of specifying a client’s user identity to the WBEM server, and then demonstrating that that client really is that particular user by specifying the user’s credentials.
- **Role assumption –** The process of assuming that a Solaris Role-Based Access Control (RBAC) role identity is to be used by the WBEM server when it checks authorization.
- **Secure messaging –** The process of adding a secure message authenticator to each client request message. This authenticator enables the WBEM server to check the origin of the message and to determine if that message was modified during its delivery to the WBEM server.
- **Authorization –** The process of verifying that an authenticated user or a role identity has been granted access to the WBEM data that is managed by each WBEM method call. You use the Solaris Management Console User tool and Sun WBEM User Manager for authorization management.

- **Auditing –** The process of writing an audit record of a specific operation that was performed by the WBEM server. These records track the changes that an authenticated user makes to the management data on the WBEM server system.
- **Logging –** The writing of particular security-related events in the WBEM log. You can view the WBEM log by using the Solaris Management Console Log Viewer.

Each mechanism is described in more detail in the sections that follow.

Authentication

When a client application connects to a CIM Object Manager server, the client's user identity must be authenticated by the CIM Object Manager on the WBEM server. The user's WBEM client must provide a Solaris user identity and its accompanied login password. The identity and credential are used in a security authentication exchange between the client and WBEM server to verify that the client is a valid Solaris user who is allowed to log in to the WBEM server system.

If the WBEM server cannot verify the user identity and credential, and the user's identity is invalid, the WBEM server returns a CIM security exception that includes the `NO_SUCH_PRINCIPAL` error.

If the WBEM server cannot verify the user's identity and credential, and the user's password is invalid for that user's identity, the WBEM server returns a CIM security exception that includes the `INVALID_CREDENTIAL` error.

Role Assumption

A role identity can be assumed only when a WBEM user selects the Remote Method Invocation (RMI) protocol. Role assumption is not supported by the XML/HTTP protocol.

The Solaris implementation of WBEM supports the ability of a client to assume the identity of a Solaris role when that client is authenticated by the CIM Object Manager on the WBEM server. When the WBEM server uses RBAC authorizations to check authorization permission, the WBEM server checks the permission that is granted to the assumed role rather than the permission that is granted to the underlying user identity.

RBAC roles are described in more detail in "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

The client must provide the Solaris role identity and password, in addition to a Solaris user identity and password when the client attempts to connect.

If the WBEM server cannot verify the Solaris role identity, the WBEM server returns a CIM security exception that includes the `NO_SUCH_ROLE` error.

If the role password is invalid for the specified role identity, the WBEM server returns the INVALID_CREDENTIAL error in the CIM security exception.

If both the role identity and role password are valid, but the user is not allowed to assume the role, the WBEM server returns the CANNOT_ASSUME_ROLE error in the CIM security exception.

Secure Messaging

In the CIM RMI protocol, each request from the client to the WBEM server contains a message authenticator that is constructed from the data parameters in the message. A one-way digest is also created with a session key established during the authentication exchange.

The WBEM server verifies this message authenticator, which guarantees that the request came from the same client that was authenticated and that the message was not modified or replayed during its communications to the server.

If the message was modified, replayed, or created by a source that was not the original client, the WBEM server returns a CIM security exception that contains the CHECKSUM_ERROR error. The WBEM server also writes a log message to the WBEM log.

Authorization

After the WBEM server connects, the WBEM server uses the authenticated user or the role identity for all authorization checks on subsequent operations with the CIM client.

WBEM supports two types of authorization checking, based on:

- Access control lists (ACLs) that are maintained by the WBEM server for specific namespaces
- RBAC authorizations that are configured as part of the Solaris operating environment

The particular authorization checking mechanism that WBEM uses depends on how the MOF class provider is implemented. The particular authorization checking mechanism that WBEM uses for a specific MOF class operation depends on:

- The particular operation that WBEM executes
- How the MOF class provider is implemented

The classes defined in Solaris_Acl1.0.mof implement WBEM ACL-based security. WBEM ACL-based security provides a default authorization scheme for Solaris WBEM Services, and, under specific circumstances, applies to a particular set of CIM operations. ACL-based security is uniquely provided by Solaris WBEM Services.

You use Sun WBEM User Manager (`wbemadmin`) to establish an ACL for a specific namespace on the WBEM server. Sun WBEM User Manager enables you to add user names, or role names, to the ACL for the namespace, and to assign each user "read" or "write" permission. Sun WBEM User Manager is described in "Using Sun WBEM User Manager to Set Access Control" on page 55 and in `wbemadmin`(1M).

Write permission allows a user to modify the class metadata, modify instances of MOF classes in that namespace, and issue an invoke method on instances. The local WBEM server `root` user identity is always granted write permission to all namespaces on the server. All authenticated users without an explicit ACL entry are granted read permission by default.

Operations that include the accessing of MOF class metadata, such as `getClass`, use the WBEM ACLs. These operations include the checking of permissions that are granted to the authenticated user by the ACL for the namespace that contains the MOF class. You can set an RBAC role in an ACL entry, but the ACL entry is always checked against the user identity rather than the role identity. In other words, you can set a role name in an ACL, but the CIM Object Manager does not check the role name at runtime.

Operations that involve MOF class instances might include the checking of either WBEM ACLs or RBAC authorizations.

You can also grant permissions to a user, or role identity, that allow that user to access and modify the instances of MOF classes whose providers use the RBAC authorizations. You grant these permissions by using the Rights tool in the Solaris Management Console User tool. The granting of permissions to a user is described in "Creating or Changing a Rights Profile" in *System Administration Guide: Security Services*.

If the instances for a MOF class are stored in the WBEM persistent datastore, the CIM Object Manager checks the WBEM ACL for the namespace that contains the MOF class. If the MOF class provider implementation accesses the provider's datastore, or accesses system data in the Solaris operating environment, the MOF class provider implementation almost always uses RBAC authorization checking.

In general, if a MOF class definition contains a Provider qualifier, the provider implementation usually makes RBAC authorization checks. If the MOF class definition does not contain a Provider qualifier, the CIM Object Manager:

- Stores the instances of that class in the WBEM persistent datastore
- Checks the ACL that controls access to the namespace for the class to ensure that access is granted

Auditing

The WBEM server writes audit records for certain events during processing. For example, the WBEM server writes audit records whenever the authentication of a client succeeds or fails, and whenever an operation that modifies user information is executed.

The WBEM server uses the underlying Solaris Basic Security Module (BSM) to write its audit records. You must enable the BSM auditing mechanism (`bsmconv`) in the Solaris operating environment on the WBEM server to ensure that audit information is recorded. This command is described in `bsmconv`(1M).

Note – If you are using Trusted Solaris™, you do *not* need to enable the BSM auditing mechanism.

Logging

The WBEM server writes log records to the WBEM log for particular security events, for example, when an authenticated session for a client is established or when authorization checking fails. You can review the WBEM log in the Solaris Management Console Log Viewer, which is described in Chapter 5.

You can identify security-related log events by the category Security log, which is listed in the Category column. You can view only security log messages by selecting the category Security in the Log Viewer filter dialog box. Most security log messages include the user identity of the client and the name of the client host.

Using Sun WBEM User Manager to Set Access Control

Sun WBEM User Manager (`wbemadmin`) enables you and other privileged users to:

- Add and delete authorized users
- Set access privileges for authorized users
- Manage user authentication and access to CIM objects on a WBEM-enabled system

Note – The user for whom you specify access control must have a Solaris user account.

What You Can and Cannot Do With Sun WBEM User Manager

You can set access privileges for individual namespaces or for a combination of a user and a namespace. When you add a user and select a namespace, the user is granted read access to CIM objects in the selected namespace by default.

Note – An effective way to combine user and namespace access rights is to first restrict access to a namespace, and then grant individual users read, read and write, or write access to that namespace.

You cannot set access rights on individual managed objects. However you can set access rights for all managed objects in a namespace as well as on a per-user basis.

If you log in as `root`, you can set the following types of access to CIM objects:

- **Read Only –** Allows read-only access to CIM Schema objects. Users with this privilege can retrieve instances and classes, but cannot create, delete, or modify CIM objects.
- **Read/Write –** Allows full read, write, and delete access to all CIM classes, instances, and invoked methods.
- **Write –** Allows write and delete access, but not read access, to all CIM classes and instances.
- **None –** Allows no access to CIM classes and instances.

Using Sun WBEM User Manager (Task Map)

The following table identifies the procedures that you need to follow to start and use Sun WBEM User Manager.

Task	Description	For Instructions
Start the Sun WBEM User Manager.	Start the Sun WBEM User Manager by using the `wbemadmin` command.	"How to Start Sun WBEM User Manager" on page 58
Grant default access rights to a user.	Grant default access rights to a user by using the Users Access tool of the Sun WBEM User Manager.	"How to Grant Default Access Rights to a User" on page 58
Change access rights for a user.	Change access rights for a user by using the Read and Write check boxes in the Sun WBEM User Manager.	"How to Change Access Rights for a User" on page 59
Remove access rights for a user.	Remove access rights for a user by using the Users Access tool of the Sun WBEM User Manager.	"How to Remove Access Rights for a User" on page 59
Set access rights for a namespace.	Set access rights for a namespace by using the Namespace Access tool of the Sun WBEM User Manager.	"How to Set Access Rights for a Namespace" on page 59
Remove access rights for a namespace.	Remove access rights for a namespace by using the Namespace Access tool of the Sun WBEM User Manager.	"How to Remove Access Rights for a Namespace" on page 60

Using Sun WBEM User Manager

This section describes how to start and use Sun WBEM User Manager.

▼ How to Start Sun WBEM User Manager

1. **Become superuser.**
2. **In a command window, type:**

   ```
   # /usr/sadm/bin/wbemadmin
   ```

   Sun WBEM User Manager starts, and a Login dialog box opens.

   Note – Context-help information is available in the Context Help panel when you click on the fields in the Login dialog box.

   ---

3. **Fill in the fields on the Login dialog box.**

   a. **In the User Name field, type the user name.**

   Note – You must have read access to the `root\security` namespace to log in. By default, Solaris users have guest privileges, which grant them read access to the default namespaces. Users with read access can view, but not change, user privileges.

   You must log in as `root` or a user with write access to the `root\security` namespace to grant access rights to users.

   ---

   b. **In the Password field, type the password for the user account.**

4. **Click OK.**

   The User Manager dialog box opens. The dialog box contains a list of users and their access rights to WBEM objects within the namespaces on the current host.

▼ How to Grant Default Access Rights to a User

1. **Start Sun WBEM User Manager.**
2. **In the Users Access portion of the dialog box, click Add.**

   A dialog box opens that lists the available namespaces.
3. **Type the name of a Solaris user account in the User Name field.**
4. **Select a namespace from the listed namespaces.**
5. **Click OK.**

   The user name is added to the User Manager dialog box.

6. **To save changes and close the User Manager dialog box, click OK. To save changes and keep the dialog box open, click Apply.**

   The user that you specified is granted read access to CIM objects in the namespace that you selected.

▼ How to Change Access Rights for a User

1. **Start Sun WBEM User Manager.**
2. **Select the user whose access rights you want to change.**
3. **To grant the user read-only access, click the Read check box. To grant the user write access, click the Write check box.**
4. **To save changes and close the User Manager dialog box, click OK. To save changes and keep the dialog box open, click Apply.**

▼ How to Remove Access Rights for a User

1. **Start Sun WBEM User Manager.**
2. **In the Users Access portion of the dialog box, select the user name for which you want to remove access rights.**
3. **Click Delete to delete the user's access rights to the namespace.**

   A confirmation dialog box opens. This dialog box prompts you to confirm your decision to delete the user's access rights.
4. **To confirm, click OK.**
5. **To save changes and close the User Manager dialog box, click OK. To save changes and keep the dialog box open, click Apply.**

▼ How to Set Access Rights for a Namespace

1. **Start Sun WBEM User Manager.**
2. **In the Namespace Access portion of the dialog box, click Add.**

   A dialog box opens. The dialog box lists the available namespaces.
3. **Select the namespace for which you want to set access rights:**

Note – By default, users have read-only access to a namespace.

- To allow no access to the namespace, make sure that the Read and Write check boxes are not selected.
- To allow write access, click the Write check box.
- To allow read access, click the Read check box.

4. **To save changes and close the User Manager dialog box, click OK. To save changes and keep the dialog box open, click Apply.**

▼ How to Remove Access Rights for a Namespace

1. **Start Sun WBEM User Manager.**
2. **In the Namespace Access portion of the dialog box, select the namespace for which you want to remove access control, and then click Delete.**

   Access control is removed from the namespace, and the namespace is removed from the list of namespaces on the Sun WBEM User Manager dialog box.
3. **To save changes and close the User Manager dialog box, click OK. To save changes and keep the dialog box open, click Apply.**

Troubleshooting Problems With WBEM Security

This section describes what to do when:

- A client (user) cannot be authenticated by the CIM Object Manager on the WBEM server
- A role cannot be assumed
- An `ACCESS_DENIED` error occurs

If a Client (User) Cannot Be Authenticated by the CIM Object Manager on the WBEM Server

If a client cannot be successfully authenticated by the CIM Object Manager on the WBEM server, the WBEM server returns a CIM security exception when it attempts to establish the CIM client handle in the client application. The exception contains an error code that indicates why the authentication attempt failed.

Error	Probable Cause	Solution
`NO_SUCH_PRINCIPAL`	Specified user identity was not valid in the Solaris operating environment on the WBEM server, or the user account for that user identity either has no password or is locked.	Check that the user has a valid user identity, that is, the user can log in to the Solaris operating environment on the WBEM server machine. The Solaris system that is set up as the WBEM server might be using user identities from a name service configured on the server, so you might need to check the name service tables.
`INVALID_CREDENTIAL`	Password for the specified user (or role, if assuming a role identity) is not valid for that user in the Solaris operating environment on the WBEM server.	Check that the user's password is correct.

Error	Probable Cause	Solution
`NO_SUCH_ROLE`	Role identity that is assumed in the authentication to the WBEM server is not a valid RBAC role in the Solaris operating environment on the WBEM server.	The role identity might be a valid entry in the `passwd` table on the server, but you will *not* be able to log in to the server under that identity (Solaris does not allow you to log in directly to role identities). So, you must check the `passwd` table for the role identity, and check the `user_attr` table to ensure that the role is defined as a role type of user. Role identities in the `user_attr` table each contain an attribute in the syntax *type=role*. You can also check for a valid user or valid role identity by using the Solaris Management Console User tool. You can use User Management to check for a user, and you can use Role Management to check for a role. However, when using the User tool, you *must* know the correct source of the tables on the CIM Object Manager server. In other words, if the CIM Object Manager server is using a name service such as NIS, you must access the master server for that name service.
`CANNOT_ASSUME_ROLE`	Role identity is valid, but the specified user identity in the authentication exchange is not configured to assume that role.	Explicitly assign users to roles by using the Administrative Role tool in the Solaris Management Console User tool collection, which is described in “Changing Role Properties” in *System Administration Guide: Security Services*.

If Other CIM Security Exception Errors Appear

The WBEM server can return other error indications in the CIM security exception. However, these indications typically identify a system failure in the authentication exchange. The WBEM client configuration might not be compatible with the WBEM server configuration for the security options in the authentication exchange.

If these error indications occur, check that the WBEM installation on the client machine contains the appropriate configuration property values for security in `WbemClient.properties`. This file is usually located in the vendor extension subdirectory in the WBEM installation directory `/usr/sadm/lib/wbem/extension`.

Also, check the client application `CLASSPATH` setting to ensure that `sunwbem.jar` and the extension directory path name are on the class path.

If an Authorization Check Fails

If a client is not authorized to access or modify the data associated with a request to the WBEM server, the WBEM server returns a CIM security exception for that request that includes the `ACCESS_DENIED` error.

The `ACCESS_DENIED` error indicates that a WBEM request could not be completed because the authenticated user or the role has not been granted the appropriate access to the data being managed by that request.

Check the security messages in the WBEM log for the failed request (viewing log data is described in "Viewing Log Data Through Log Viewer" on page 65). Authorization failure log messages specify `Access denied` in the Summary column. The User column lists the name of the authenticated user or the role name that was used in the check. The Source column lists the name of the provider that is making the check. Note that the name of the provider that is listed in this column is a user-friendly provider name, not the provider implementation class name.

The detailed message contains the name of the permission that was being checked, and that has not been granted to the user or role.

If the permission appears as *namespace*:*right*, the authorization check was using a namespace ACL. The authenticated user has not been granted that permission (read or write) for that namespace.

Use Sun WBEM User Manager (`wbemadmin`) to grant the user the appropriate permission. Sun WBEM User Manager is described in "Using Sun WBEM User Manager to Set Access Control" on page 55.

If the permission appears as `solaris.`*application*`.`*right*, the authorization check was using an RBAC authorization.

Use the Administrative Role tool in the Solaris Management Console User tool collection to grant the rights that you want to the user or role. This procedure is described in “Changing Role Properties” in *System Administration Guide: Security Services*.

CHAPTER **5**

Viewing System Log Data (Tasks)

WBEM log files enable system administrators to track errors, warnings, and informational messages that the management subsystem generates. The WBEM logging service enables application developers and writers of providers to write log messages to the log files. For example, you might want to write out log messages when a system is not able to access a serial port, when a system successfully mounts a file system, or when the number of processes that are running on a system exceeds the allowed number.

This chapter describes how to view log data.

Viewing Log Data Through Log Viewer

After you have created a log record, you can start the Solaris Management Console application and Log Viewer. A log record is automatically created when you start the Solaris Management Console software.

You can view all details of a log record in the Solaris Management Console Log Viewer.

▼ How to Start the Solaris Management Console Application and Log Viewer

1. **Type this command:**

```
$ smc
```

2. **In the Navigation panel, either double-click This Computer or single-click the expand/compress icon next to This Computer.**

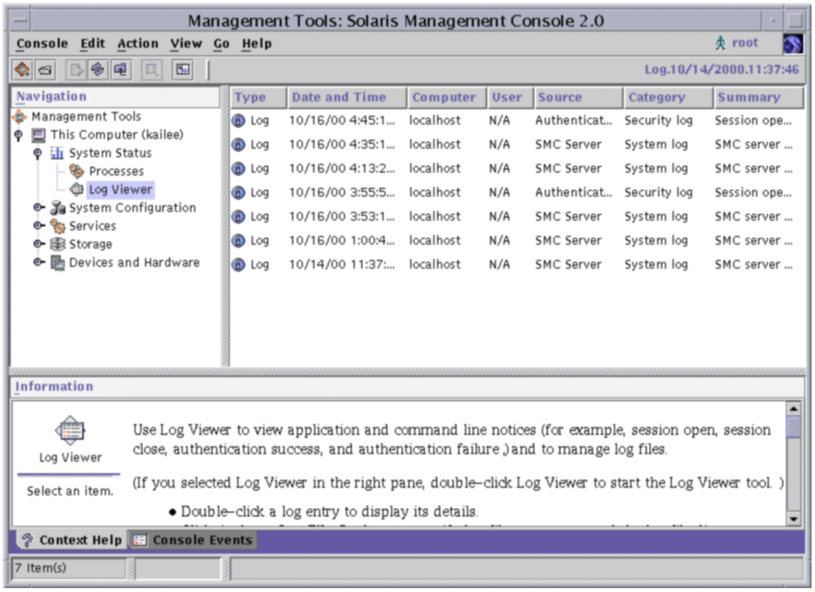
A tree of commands is displayed below This Computer.

3. **Double-click System Status.**

   The Log Viewer icon is displayed.

4. **Click the Log Viewer icon.**

   Log Viewer starts.



Type	Date and Time	Computer	User	Source	Category	Summary
Log	10/16/00 4:45:1...	localhost	N/A	Authenticat...	Security log	Session ope...
Log	10/16/00 4:35:1...	localhost	N/A	SMC Server	System log	SMC server ...
Log	10/16/00 4:13:2...	localhost	N/A	SMC Server	System log	SMC server ...
Log	10/16/00 3:55:5...	localhost	N/A	Authenticat...	Security log	Session ope...
Log	10/16/00 3:53:1...	localhost	N/A	SMC Server	System log	SMC server ...
Log	10/16/00 1:00:4...	localhost	N/A	SMC Server	System log	SMC server ...
Log	10/14/00 11:37:...	localhost	N/A	SMC Server	System log	SMC server ...

Use Log Viewer to view application and command line notices (for example, session open, session close, authentication success, and authentication failure.) and to manage log files.

(If you selected Log Viewer in the right pane, double–click Log Viewer to start the Log Viewer tool.)

- Double–click a log entry to display its details.

FIGURE 5–1 Solaris Management Console Application, Log Viewer Selected

Index