BEA WebLogic Server™

WebLogic SNMP Management Guide

Release 7.0
Document Revised: August 28, 2002

Copyright

Restricted Rights Legend

Trademarks or Service Marks

WebLogic SNMP Management Guide

Part Number	Document Revised	Software Version
N/A	August 28, 2002	BEA WebLogic Server Version 7.0

Contents

About This Document

This document explains the management subsystem provided for configuring and monitoring your WebLogic Server implementation. It covers the following topics:

- Chapter 1, "Using SNMP to Manage WebLogic Server," describes basic concepts of Simple Network Management Protocol as they apply to managing WebLogic Servers. Setting up the WebLogic SNMP agent is also described.
- Chapter 2, "Trap Notifications," describes the characteristics of WebLogic enterprise-specific SNMP trap notifications and how to configure the WebLogic SNMP agent to generate SNMP traps.
- Chapter 3, "Using Multiple SNMP Agents," describes how to use the WebLogic SNMP agent as a master agent that proxies for other SNMP agents.

Audience

This document is intended mainly for system administrators who will be managing the WebLogic Server application platform and its various subsystems.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

How to Print the Document

You can print a copy of this document from a Web browser, one main topic at a time, by using the File—Print option on your Web browser.

A PDF version of this document is available on the WebLogic Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at http://www.adobe.com.

Contact Us!

Your feedback on BEA documentation is important to us. Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the documentation.

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at http://www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using

- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Usage
Ctrl+Tab	Keys you press simultaneously.
italics	Emphasis and book titles.

`monospace text`	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that the user is told to enter from the keyboard.
	Examples:
	`import java.util.Enumeration;`
	`chmod u+w *`
	`config/examples/applications`
	`.java`
	`config.xml`
	`float`

`monospace italic text`	Placeholders. *Example*: `String CustomerName;`

`UPPERCASE MONOSPACE TEXT`	Device names, environment variables, and logical operators.
	Examples:
	`LPT1`
	`BEA_HOME`
	`OR`

`{ }`	A set of choices in a syntax line.

Convention	Usage
`[ ]`	Optional items in a syntax line. *Example*: `java utils.MulticastTest -n` *`name`* `-a` *`address`* `[-p` *`portnumber`*`] [-t` *`timeout`*`] [-s` *`send`*`]`

`|`	Separates mutually exclusive choices in a syntax line. *Example*:
	`java weblogic.deploy [list|deploy|undeploy|update]`
	`password {application} {source}`

`...`	Indicates one of the following in a command line: ■ An argument can be repeated several times in the command line. ■ The statement omits additional optional arguments. ■ You can enter additional parameters, values, or other information
. . .	Indicates the omission of items from a code example or from a syntax line.

1 Using SNMP to Manage WebLogic Server

WebLogic Server software includes the ability to communicate with enterprise-wide management systems using Simple Network Management Protocol (SNMP). The WebLogic Server SNMP capability enables you to integrate management of WebLogic Servers into an SNMP-compliant management system that gives you a single view of the various software and hardware resources of a complex, distributed system.

This section discusses the following topics:

The SNMP Agent/Manager Model

SNMP management is based on the agent/manager model described in the network management standards defined by the International Organization for Standardization (ISO). In this model, a network/systems manager exchanges monitoring and control information about system and network resources with distributed software processes called **agents**.

Any system or network resource that is manageable through the exchange of information is a **managed resource**. This could be a software resource such as a Java Database Connectivity (JDBC) connection pool or a hardware resource such as a router.

Agents function as "collection devices" that typically gather and send data about the managed resource in response to a request from a manager. In addition, agents often have the ability to issue unsolicited reports to managers when they detect certain predefined thresholds or conditions on a managed resource. In SNMP terminology, these unsolicited event reports are called **trap notifications**.

A manager relies upon a database of definitions and information about the properties of managed resources and the services the agents support — this makes up the Management Information Base (MIB). When new agents are added to extend the management reach of a manager, the manager must be provided with a new MIB component that defines the manageable features of the resources managed through that agent. The manageable attributes of resources, as defined in an SNMP-compliant MIB, are called **managed objects**. Defining the heterogeneous components of an enterprise's distributed systems within a common MIB on the management station provides a unified perspective and single access point for managing system and network resources.

The SNMP Agent Role in a WebLogic Domain

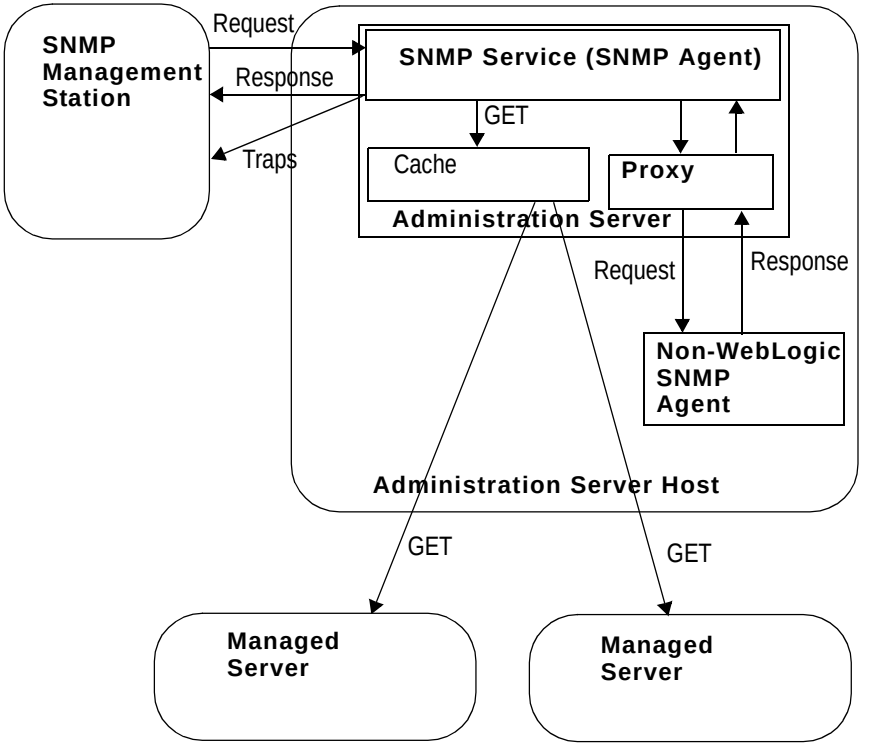
An inter-related set of WebLogic Server resources managed as a unit is called a **domain**. A domain includes one or more WebLogic Servers, and may include WebLogic Server clusters.

Within each WebLogic domain one server is the Administration Server; other servers in the domain are Managed Servers. A typical J2EE application may include components distributed across multiple Managed Servers. The Administration Server provides the central point of control for configuring and monitoring the entire WebLogic domain. For more on WebLogic domains, see the WebLogic Server Administration Guide.

The WebLogic Administration Server also has the ability to run the SNMP Service. When the SNMP Service is enabled for a WebLogic domain, the Administration Server is functioning as the SNMP agent for that WebLogic domain. (See Figure 1-1.)

For information on enabling and configuring the WebLogic SNMP Service, refer to "Setting Up the WebLogic SNMP Agent" in the Administration Console Online Help.

Figure 1-1 SNMP Management of a WebLogic Domain



You can use the WebLogic SNMP agent to do the following:

- Respond to simple GET requests from an SNMP manager for the current value of WebLogic attributes.
- Send trap notifications to SNMP managers when the Administration Server comes up and when any Managed Server goes up or down.

- Send trap notifications to SNMP managers when messages are logged in a Managed Server that satisfy criteria that you specify.
- Send trap notifications to SNMP managers when a WebLogic configuration attribute that you specify has changed value.
- Offload polling of WebLogic attributes to the WebLogic Administration Server using standard JMX monitors, based on thresholds and polling intervals that you define. A trap notification is sent to the SNMP manager when the criteria you specify are satisfied.
- Act as a proxy agent that passes requests from an SNMP manager to other SNMP agents (such as an Oracle database agent) on the same machine.

WebLogic Server Managed Resources and MBeans

Resources on WebLogic Server instances use Java Management Extensions (JMX) Managed Beans (MBeans) to expose their management functions. An **MBean** is a concrete Java class that is developed per JMX specifications. It can provide getter and setter operations for each management attribute within a managed resource along with additional management operations that the resource makes available.

When you configure the WebLogic SNMP agent to collect information from managed resources, you must specify the name of the MBean and MBean attribute from which you want to collect data.

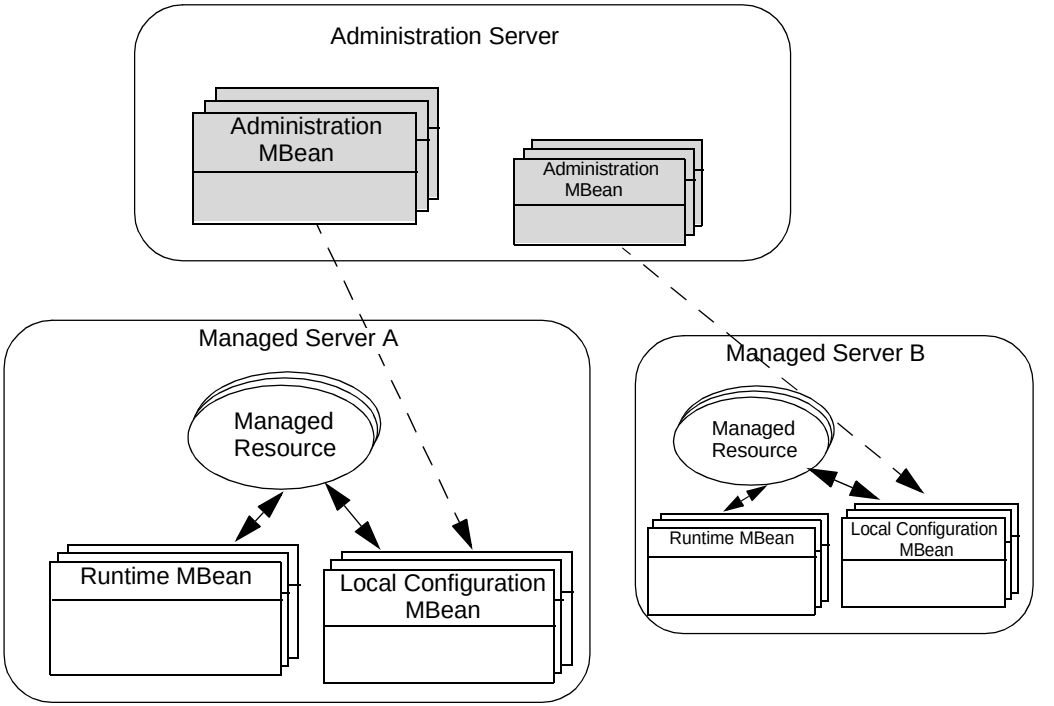
WebLogic Server MBeans that expose the configuration data of a managed resource are called **Configuration MBeans** while MBeans that provide performance metrics and other information about the runtime state of a managed resource are called **Runtime MBeans**. For example, a `ServerMBean` Configuration MBean indicates the listen port for a server instance while the `ServerRuntimeMBean` Runtime MBean indicates the current lifecycle state of a server instance.

Each WebLogic Server instance hosts its own set of Runtime MBeans to report its performance metrics and runtime state.

However, to support the WebLogic Server model of centralizing management responsibilities onto the Administration Server, the Administration Server hosts Configuration MBeans for all managed resources on all server instances in the domain. In addition, each server instance hosts a local replica of its Configuration MBean that is on the Administration Server. The Configuration MBeans on the Administration Server are called **Administration MBeans**, and the replicas on the Managed Servers are called **Local Configuration MBeans**. (See Figure 1-2.)

This distribution of Configuration MBeans creates one centralized point of management for all server instances in the domain while maximizing performance.

Figure 1-2 Distribution of MBeans



Note that managed resources on server instances use the Local Configuration MBean replicas instead of initiating remote calls to the Administration Server. Because it is possible to override the configuration of an Administration MBean and directly set a value in a Local Configuration MBean, it is possible that an Administration MBean does not report the attribute value that a local managed resource is currently using. For

example, assume that ManagedServerA is configured to use a listen port of `7001`. Through the `weblogic.Server` startup command, a JMX API, or the `weblogic.Admin` utility, you temporarily set the listen port value for an of ManagedServerA to `8001`. The Administration MBean will report a listen port of `7001`, but the Local Configuration MBean will correctly report the currently used value of `8001`.

For more information about MBeans on WebLogic Server, refer to the following:

- "Documentation for Configuration MBean APIs" on page 1-6
- "Documentation for Runtime MBean APIs" on page 1-7
- "Overview of WebLogic JMX Services" in the *Programming WebLogic Management Services with JMX* guide

Documentation for Configuration MBean APIs

To view the documentation for Configuration MBeans, do the following:

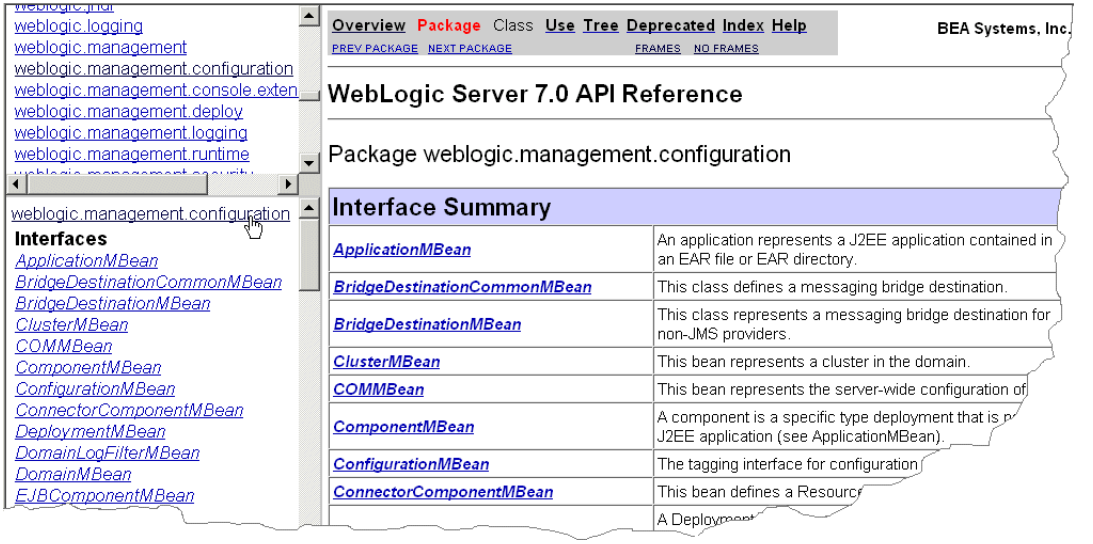
1. Open the WebLogic Server Javadoc.
2. In the top left pane of the Web browser, click `weblogic.management.configuration`.

   The lower left pane displays links for the package.

3. In the lower left pane, click `weblogic.management.configuration` again.

   The right pane displays the package summary. (See Figure 1-3.)

Figure 1-3 Javadoc for the configuration Package



4. Click on an interface name to view its API documentation.

Documentation for Runtime MBean APIs

To view the documentation for Runtime MBeans, do the following:

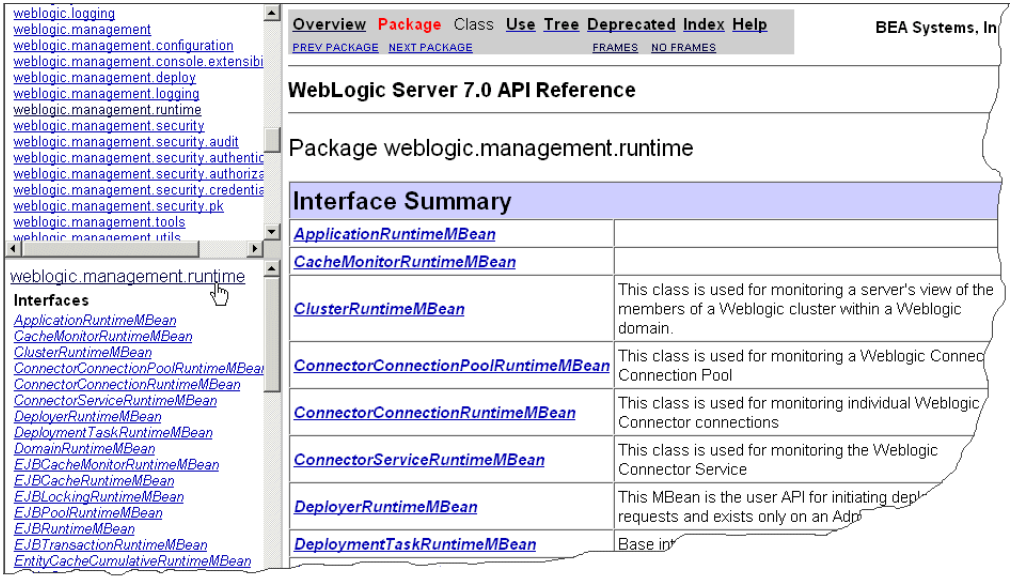
1. Open the WebLogic Server Javadoc.
2. In the top left pane of the Web browser, click `weblogic.management.runtime`.

   The lower left pane displays links for the package.

3. In the lower left pane, click `weblogic.management.runtime` again.

   The right pane displays the package summary. (See Figure 1-4.)

Figure 1-4 Javadoc for the runtime Package



4. Click on an interface name to view its API documentation.

SNMP MIB for WebLogic

The WebLogic Server MIB assigns a unique number called an **object identifier** (OID) to its MBean attributes. Each MBean attribute in the MIB is an SNMP managed object and is manageable by an SNMP management system.

The MIB creates a hierarchical relationship between managed objects and expresses the hierarchy in a tree structure, called the MIB tree or registration tree. Each OID in the MIB consists of a left-to-right sequence of integers. This sequence defines the location of the object in the MIB tree. By specifying a unique path through the tree to the object, the OID allows the object to be identified uniquely. Each node in the path

defined in an OID has both a number and a name associated with it. The path `.1.3.6.1.4.1` defines the `private.enterprises` OID and each number beneath that node on the tree represents the branches in the tree reserved for a particular vendor.

The BEA MIBS are registered at the location `.1.3.6.1.4.1.140` in the tree. And the WebLogic Server MIB consists of all OIDs below `.1.3.6.1.4.140.625`. For example, `.1.3.6.1.4.1.140.625.360.1.60` is the OID for `serverRuntimeState`. When an SNMP manager requests the current value of a WebLogic attribute, it indicates the attribute by specifying the corresponding OID.

The MIB is located in a file named `WL_HOME\server\lib\BEA-WEBLOGIC-MIB.asn1`. For more information about the contents of the WebLogic MIB, refer to the WebLogic Server SNMP MIB Reference.

Note: The enterprise OID for the SNMP agent for WebLogic 6.1 differs from the enterprise OID for WebLogic used with the WebLogic 5.1 SNMP agent. The enterprise OID for WebLogic 6.1 is `.1.3.6.1.4.140.625`. The textual name of the WebLogic Server node in the registration tree is now `wls`.

SNMP Community Names

To ensure that the entity requesting data from the WebLogic SNMP agent has permission to obtain the data, and to verify that the agent has permission to send trap notifications to a target manager, SNMP uses textual passwords called **community names**.

When you set up the SNMP agent capability of the WebLogic Administration Server (described in "Setting Up the WebLogic SNMP Agent" in the Administration Console Online Help), one of the things you must specify is the community name that the agent expects from the SNMP manager. If the agent receives an SNMP request with an incorrect community name, it automatically generates an `authenticationFailure` trap that is sent to the source of the request.

Specifying the Target Server in Management Requests

In order to retrieve any attribute of any WebLogic Server in a domain, the SNMP manager must send the request to the Administration Server, acting as the SNMP agent for the domain. Because the same attribute, such as `serverUptime`, may exist on each of the WebLogic Servers, there needs to be a way to determine which WebLogic Server the manager is requesting data for.

In order to accomplish this, the target server name is appended to the SNMP password (community) as part of the community string sent with that request.

To specify a particular Managed Server, the SNMP manager sends to the WebLogic SNMP agent a community string that has the following form:

`community_prefix@server_name`

where *`community_prefix`* is the actual SNMP community name and *`server_name`* is the name of the target Managed Server. The *`community_prefix`* value sent by the manager must match the value that you set in the Community Prefix field when you configure the SNMP agent (as described in "Setting Up the WebLogic SNMP Agent" in the Administration Console Online Help).

To send a management request for the attributes of the Administration Server, the SNMP manager must send a community string to the WebLogic SNMP agent that has the following form:

`community_prefix`

where *`community_prefix`* is the actual SNMP community name. The *`community_prefix`* value sent by the manager must match the value that you set in the Community Prefix field when you configure the SNMP agent (as described in "Setting Up the WebLogic SNMP Agent" in the Administration Console Online Help).

If the SNMP manager sends a community string with the form

`community_prefix@domain_name`

the Administration Server returns the values of the specified attribute for every server in the domain.

How to Access Runtime Information

Note: Consult the vendor information for specific information on how to configure your SNMP Management system to access runtime information.

This section provides an example of how to obtain runtime information on the `serverRuntimeListenAddress` for an Administration Server (myserver) and a Managed Server (clusterServer1). The MIB definition for `serverRuntimeListenAddress` is:

```
iso(1).org(3).dod(6).internet(1).private(4).enterprises(1)
.bea(140).wls(625).serverRuntimeTable(360).serverRuntimeEntry(1)
.serverRuntimeListenAddress(30)
```

The following examples use a generic tool, `snmpgetnext`, to provide runtime information. `snmpgetnext` has the following arguments:

Argument	Description
`-c` *`communitystring`*	Specifies the community string sent to the WebLogic Server SNMP agent.
`host`	Name of the host.
`OID`	Object Identifier for the runtime information you wish to access.

To get the `serverRuntimeListenAddress` for a Managed Server:

```
snmpgetnext  -c public@clusterServer1 localhost .1.3.6.1.4.1.140.625.360.1.30
```

To get the `serverRuntimeListenAddress` for an Administration Server, use one of the following methods:

```
snmpgetnext  -c public@myserver      localhost  .1.3.6.1.4.1.140.625.360.1.30
snmpgetnext  -c public               localhost  .1.3.6.1.4.1.140.625.360.1.30
snmpgetnext                          localhost  .1.3.6.1.4.1.140.625.360.1.30
```

CHAPTER

2 Trap Notifications

You can configure the WebLogic SNMP agent to detect certain thresholds or conditions within a managed resource and send a report (trap notification) to one or more SNMP managers.

This topic describes the trap notifications that the WebLogic SNMP agent can generate:

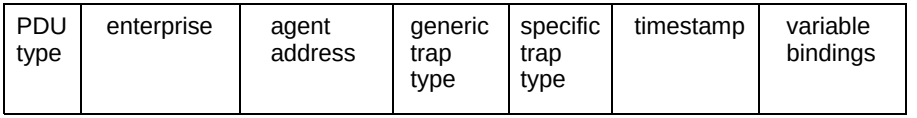
This topic also describes how to disable trap notifications, on "Disabling Trap Generation" on page 2-11.

For information on specifying the destinations of trap notifications from the WebLogic SNMP agent, refer to "Using SNMP to Manage WebLogic Server" on page 1-1.

Format of WebLogic Trap Notifications

The WebLogic SNMP agent sends each trap notification to SNMP managers in the form of a protocol data unit (PDU) with the fields indicated in Figure 2-1.

Figure 2-1 SNMP Trap Packet



The fields have the following meaning:

- `PDU type` identifies the packet as a trap notification.
- `enterprise` is the vendor identification (OID) for the systems/network management subsystem that generated the trap. All traps generated by the WebLogic SNMP agent have the WebLogic OID `.1.3.6.1.4.140.625` in the `enterprise` field.
- `agent address` is the IP address of the WebLogic Server instance on which the trap was generated.
- `generic trap type` is an integer in the range of 0 to 6. Table 2-2 lists the values that the different types of WebLogic SNMP traps supply for the `generic trap type` field.

Table 2-1 Values for the Generic Trap Type Field

WebLogic Trap	**Generated When**	`generic trap type` **Value**
`coldStart`	The Administration Server starts.	0
`authenticationFailure`	An SNMP manager sends an incorrect community string. The community string prefix is the actual password and must match the value that you set in the Community Prefix field of the Administration Console. (See Using SNMP to Manage WebLogic Server.)	4
All other WebLogic SNMP traps		6

Traps with a `generic trap value` of 6 are called *enterpriseSpecific* traps and are accompanied by a value in the `specific trap type` field.

- `specific trap type` is a number that further qualifies an enterpriseSpecific trap. Table 2-2 lists the values that the different types of WebLogic SNMP traps supply for the `specific trap type` field.

Table 2-2 Values for the Specific Trap Type Field

WebLogic Trap	Generated When	`specific trap type` Value
All Log Message Traps	A server instance logs a message that matches user-defined criteria for a sending a log notification trap.	60
`serverStart` Trap	A Managed Server that was down is now up.	65
`serverShutDown` Trap	A Managed Server that was up is now down.	70
All Monitor Traps	A user-defined JMX monitor detects the crossing of a threshold or occurrence of an event.	75
All Attribute Change Trap	An attribute selected by the user has changed in value.	80

- `timestamp` is the length of time between the last re-initialization of the WebLogic SNMP agent and the time at which the trap was issued.
- `variable bindings` consists of name/value pairs that further describe the trap notification. Subsequent sections in this topic describe the name/value pairs for each type of trap notification:
  - "Automatically Generated WebLogic SNMP Traps" on page 2-4
  - "Variable Bindings in Log Message Traps" on page 2-6
  - "Variable Bindings in Monitor Traps" on page 2-9
  - "Variable Bindings in Attribute Change Traps" on page 2-10

Automatically Generated WebLogic SNMP Traps

If you enable the SNMP service for a domain, the WebLogic SNMP agent automatically generates the trap notifications described in Table 2-3. Some of these traps include name/value pairs in the PDU to further describe the event.

Table 2-3 Automatically Generated Trap Notifications

Trap	Generated When	Variable Bindings
`coldStart`	The Administration Server starts.	none
`authenticationFailure`	An SNMP manager sends an incorrect community string. The community string prefix is the actual password and must match the value that you set in the Community Prefix field of the Administration Console. (See Using SNMP to Manage WebLogic Server.)	none
`serverStart`	A WebLogic Managed Server that was down is now up.	Contains two name/value pairs to identify server start time and the server name.
`serverShutDown`	A Managed Server that was up is now down.	Contains two name/value pairs to identify server down time and the server name.

Log Message Traps

Subsystems and deployable modules (such as applications) on a WebLogic Server instance generate log messages to communicate status or other operational data.

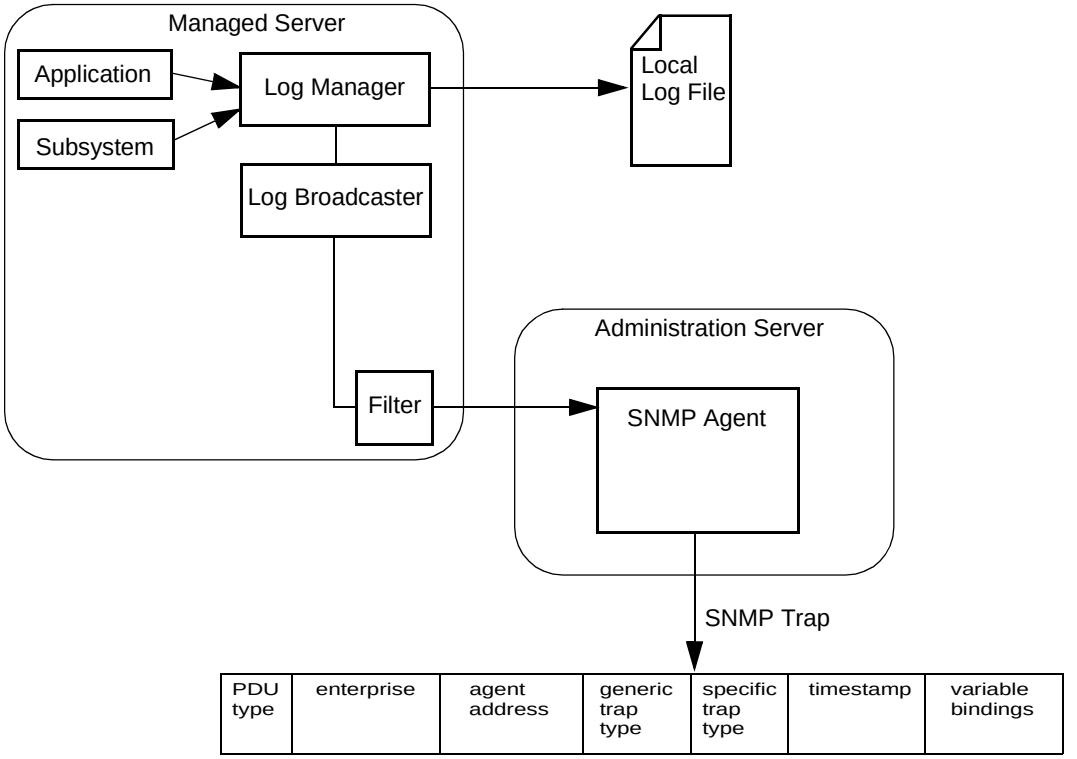
Each server instance saves these messages in a local log file and then broadcasts them as JMX notifications. You can set up the WebLogic SNMP agent to listen for all of these messages or you can set up a filter based on criteria such as the following:

- The severity level of the message
- The name of the subsystem that generated the message
- The user ID under which the subsystem is running
- A unique message ID
- A string within the message text

For example, you can specify that only messages from the Security Service of severity level `ERROR` or higher are sent to the SNMP agent. For information on setting up the SNMP agent to listen for messages, refer to "Create a Notification Log Filter" in the *Administration Console Online Help*.

When the agent receives a message, it generates an SNMP log notification trap. (See Figure 2-2.)

Figure 2-2 Log Message Traps



Variable Bindings in Log Message Traps

This section describes the name/value pairs that the log messages traps pass to the SNMP manager in the variable bindings field:

- `trapTime` — Time when the trap is generated.
- `trapServerName` — Name of the server instance on which the log message was generated.
- `trapMachineName` — Name of the machine on which the server instance is running.

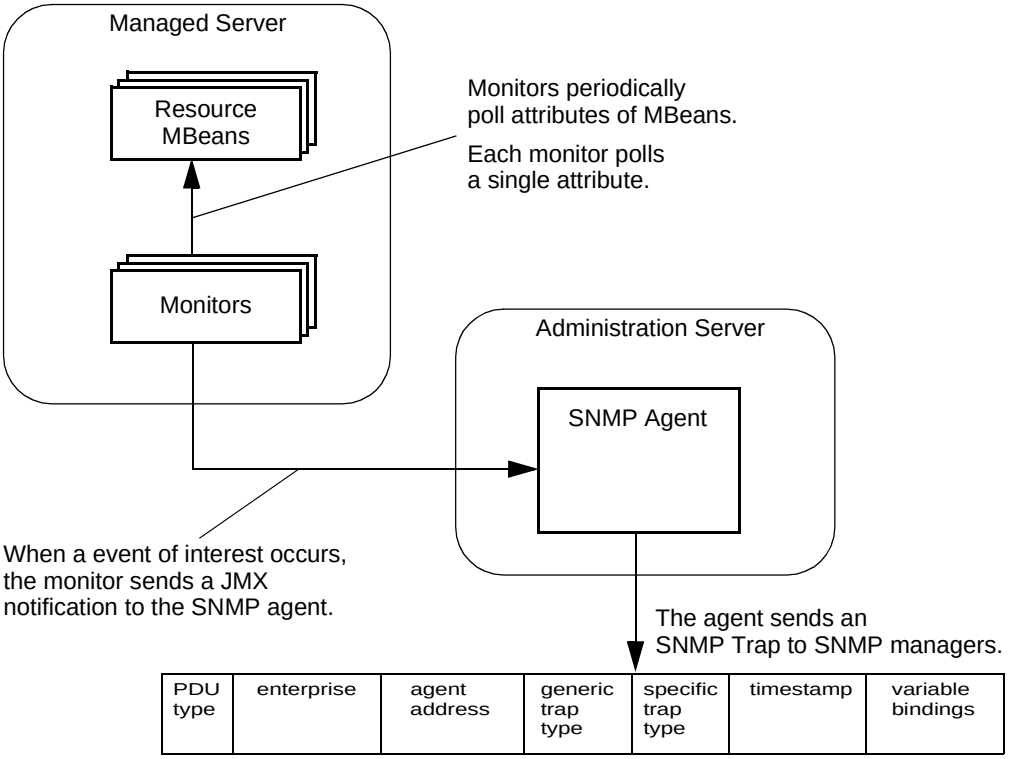
- `trapLogThreadId` — Thread ID from the log message.
- `trapLogTransactionId` — Transaction ID, if any, from the log message. Transaction ID is present only for messages logged within the context of a transaction.
- `trapLogUserId` — The user ID from the log message. The user ID indicates the the security context in which the log message was generated.
- `trapLogSubsystem` — The subsystem that generated the log message.
- `trapLogMsgId` — The log message ID from the log message.
- `trapLogSeverity` — The message severity level from the log message.
- `trapLogMessage` — The text of the log message.

For more information on log messages and the WebLogic Server logging subsystem, refer to "Using Log Messages to Manage WebLogic Server" in the *WebLogic Server Administration Guide*.

Monitor Traps

Instead of using an SNMP manager to periodically poll WebLogic resources for changes in conditions, you can configure Java Management Extension (JMX) monitors and assign them to instances of WebLogic Server. The monitors poll the MBeans of WebLogic resources at a specified interval and send notifications to the WebLogic SNMP agent when an event that you specify occurs, such as the crossing of a threshold. The SNMP agent generates a trap notification and sends it to the SNMP managers. (See Figure 2-3.)

Figure 2-3 Monitor Traps



If you are unfamiliar with WebLogic Server MBeans and how they are distributed throughout a WebLogic domain, refer to "WebLogic Server Managed Resources and MBeans" on page 1-4.

You can configure three types of JMX monitors, depending on the data type of the attribute that you want to observe (the MBean's Javadoc describes the type of data that its attributes return):

- **Counter Monitor**

  A counter monitor observes attribute values that are returned as an `Integer` object type.

  You can specify that a trap is generated if an attribute is beyond the bounds of a threshold value. You can also specify that if a value exceeds a threshold, the monitor increases the threshold by an offset value. Each time the observed

attribute exceeds the new threshold, the threshold is increased by the offset value, up to a maximum allowable threshold that you specify.

For information on configuring a counter monitor, refer to "Create a Counter Monitor" in the *Administration Console Online Help*.

- **Gauge Monitor**
  A gauge monitor observes changes in MBean attributes that are expressed as integers or floating-point.

  You can specify that a trap is generated if an attribute is beyond the bounds of a high or low threshold value.

  For information on configuring a gauge monitor, refer to "Create a Gauge Monitor" in the *Administration Console Online Help*.

- **String Monitor**
  A string monitor observes changes in attributes that are expressed as `String` objects.

  You can specify that a trap is generated if there is a match between the value and the string you provide, or you can specify that the trap is generated if the value differs from the string you provide.

  For information on configuring a string monitor, refer to "Create a String Monitor" in the *Administration Console Online Help*.

Variable Bindings in Monitor Traps

A JMX monitor polls for a specified threshold or condition and the agent generates a monitor trap when the specified threshold is crossed, or the specified condition occurs. The WebLogic SNMP agent includes the following name/value pairs in the variable bindings of each monitor trap:

- `trapTime` — The time at which the trap was generated.
- `trapServerName` — The local server whose attribute value generated the trap.
- `trapMonitorType` — Either `CounterMonitor`, `StringMonitor`, or `GaugeMonitor`.

- `trapMonitorThreshold` — An ASCII representation of the threshold that triggered the trap.
- `trapMonitorValue` — An ASCII representation of the value that triggered the trap.
- `trapMBeanName` — The name of the MBean that contained the attribute being monitored.
- `trapMBeanType` — The type of the MBean that contained the attribute being monitored.
- `trapAttributeName` — The name of the attribute whose value triggered the trap.

Attribute Change Traps

While you can use JMX monitors to periodically poll WebLogic Server resources for changes to attributes that exceed the bounds of specific thresholds, you can also configure the SNMP agent to send a trap immediately after an attribute is changed in any way. For example, you can use a monitor to poll for changes in the current number of active JDBC pools. If the number of active pools exceeds a threshold, the SNMP agent can send a trap. You would use an attribute change trap to detect whether an attribute such as the name of a JDBC pool or the listen port has been changed.

For information on configuring the SNMP agent to send attribute change traps, refer to "Create an Attribute Change" in the Administration Console Online Help.

Variable Bindings in Attribute Change Traps

An attribute change trap notification includes the following name/value pairs in the variable bindings:

- `trapTime` — The time at which the trap was generated.
- `trapServerName` — The name of the Administration Server.
- `trapMBeanName` — Name of the MBean that includes the attribute.

- `trapMBeanType` — Type of the MBean that includes the attribute.
- `trapAttributeName` — Name of the configuration attribute that has changed.
- `trapAttributeChangeType` — The value can be either `ADD`, `REMOVE`, or `UPDATE`.
- `trapAttriruteOldVal` — Value of the attribute before the change.
- `trapAttributeNewVal` — Value of the attribute after the change.

Note: Creation of monitors for changes in run-time attributes is not supported. Only attributes in the configuration MIB can be monitored for change of attribute value.

Disabling Trap Generation

When you create an entry for a particular type of trap such as a log filter trap or JMX monitor trap, generation of such traps is only activated once the Administration Server is restarted. However, for any trap request that you have created, you can de-activate the trap generation dynamically via the Administration Console (or the `weblogic.Admin` command line interface).

When you enable trap generation for a particular type of trap, you create an entry in the table for that type of trap that is displayed in the Administration Console. To de-activate that trap, simply delete the entry in the trap table. Thus, if you have created a JMX counter monitor to poll for a specified condition, you can turn off that monitor by deleting the entry for that counter monitor in the table at SNMP→Traps→Monitors→SNMP Counter Monitors.

3 Using Multiple SNMP Agents

This section discusses the following topics:

The original SNMP management model allowed for only a single, monolithic agent to carry out all management responsibilities on a given network node (IP address). This solution was not flexible enough to provide for effective management of increasingly complex systems. In addition to the agents typically provided by computer manufacturers for hardware and operating system information, agents are also produced by vendors of other products, such as agents for SQL database systems. Complex and heterogeneous systems thus require the ability to accommodate multiple agents on a single network node.

SNMP Agent as Proxy for Other Agents

This weakness of the original SNMP model led to the concept of an SNMP master agent that acts as a proxy for other SNMP agents. The WebLogic SNMP agent can function as a master agent in this sense. To use the master agent functionality of the WebLogic SNMP agent, you can assign branches of the registration tree (OID tree) as the responsibility of other SNMP agents. Each of these will be a branch that encompasses the private MIB (or some part of that MIB) which the target agent is designed to manage.

The WebLogic SNMP agent listens for requests from SNMP managers and then fans out these requests to other SNMP agents on the Administration Server machine, if the attribute requested has an OID falling under the branch of the OID tree assigned to one of those other agents. By default the WebLogic SNMP agent listens for management requests on port 161. If the WebLogic SNMP agent is to proxy for other SNMP agents, then those other agents must be configured to listen for SNMP management requests on a port other than the port that the WebLogic SNMP agent is using to receive requests from SNMP managers.

The Microsoft Windows SNMP Service

While the WebLogic Server SNMP agent can be a proxy for other SNMP agents, it cannot be configured as a subagent of the Microsoft Windows SNMP agent service.

Using Microsoft Extension Agent API, the Microsoft Windows 2000 SNMP agent service can be a proxy for other SNMP agents. However, WebLogic Server does not support this feature and cannot use the Windows SNMP agent as a proxy.

Configuring an SNMP Proxy

To configure the WebLogic SNMP agent to proxy for another SNMP agent, do the following:

1. Invoke the Administration Console (if it isn't running already).
2. Select SNMP—SNMP Proxies in the left pane. This invokes the SNMP Proxies table, which lists entries for all the SNMP agents you have configured the WebLogic SNMP agent to proxy for.
3. To create a new proxy, select the Create a new SNMP Proxy link to invoke the SNMP Proxy configuration screen. Fill out the fields on this screen as follows:
   - Name — Enter a name for the proxy in this field. This should be descriptive of the agent that the requests will be forwarded to, such as "OracleDBAgent."

- Port — Enter a port number for communication with the other SNMP agent. The agent being proxied for must be configured to expect SNMP management requests on this port number. This must be a port number other than the port being used by the WebLogic SNMP agent for communication with SNMP managers.
- OID Root — This is an absolute OID that designates the root, or top node, of the part of the OID tree being assigned to that agent.
- Community — This is the community name that the other agent expects in requests from SNMP managers.
- Timeout — This is the interval, in seconds, that the WebLogic SNMP proxy agent waits for a response to requests forwarded to another SNMP agent. If this interval elapses without a response from the other agent, the WebLogic SNMP agent will send an appropriate error to the requesting manager.

4. Click Apply to create the new proxy.
5. Restart the Administration Server so that your changes can take effect.

A Sources of SNMP Information

This appendix lists sources of additional information about Simple Network Management Protocol, including the following:

Reference Books

If you need additional information about MIBs, agents, or the SNMP protocol, refer to these books:

- Comer, Douglas; Internetworking with TCP/IP, Vol. 2; Prentice-Hall, Englewood Cliffs, New Jersey, 1991
- Leinwand, Allan and Fang, Karen; Network Management: A Practical Perspective; Addison-Wesley, Reading, Massachusetts, 1993
- Rose, Marshall T.; The Simple Book: An Introduction to Management of TCP/IP-based Internets; Prentice-Hall, Englewood Cliffs, New Jersey, 1991
- Rose, Marshall T.; The Open Book: A Practical Perspective on Open Systems Interconnection; Prentice-Hall, Englewood Cliffs, New Jersey, 1989

- Miller, Mark; Managing Internetworks with SNMP, M & T Books
- Stallings, William; SNMP, SNMPv2 and CMIP: The Practical Guide to Network Management Standards, Addison-Wesley, Reading, Massachusetts, 1993

Standards and Drafts

The SNMP protocol has been defined through a series of Requests for Comments (RFCs). The following standards and drafts are available.

Figure 3-1 SNMP RFCs

RFC Number	Description
052	IAB Recommendations
1089	SNMP over Ethernet
1109	Ad-hoc Review
1155	Structure of Management Information
1156	Management Information Base (MIB-I)
1157	SNMP Protocol
1161	SNMP over OSI
1187	Bulk table retrieval
1212	Concise MIB definitions
1213	Management Information Base (MIB-II)
1214	OSI MIB
1215	Traps
1227	SNMP Multiplex (SMUX)

RFC Number	Description
1228	SNMP-DPI
1229	Generic-interface MIB extensions
1230 IEEE 802.4	Token Bus MIB
1231 IEEE 802.5	Token Ring MIB
1239	Reassignment of MIBs
1243	AppleTalk MIB
1248	OSPF MIB
ISO 8824	ASN.1
ISO 8825	BER for ASN.1

Obtaining RFCs

You can obtain Requests for Comments in the following ways:

- Download them from almost anywhere on the Internet
- Obtain them from SRI International

  Mailing Address: SRI International, EJ291, DDN Network Information Center, 333 Ravenswood Ave., Menlo Park CA 94025

  Phone: +1.800.235.3155

  e-mail: MAIL-SERVER@nisc.sri.com. Leave the subject field blank. In the body, enter: SEND RFCnnnn.TXT-1

  FTP: ftp://ftp.nisc.sri.com/rfc/rfcNNNN.txt

Index