BEAWebLogic
Server®

WebLogic SNMP
Management Guide

Copyright

Copyright © 1995-2005 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

Trademarks and Service Marks

Contents

4. Understanding SNMP Proxies

Introduction and Roadmap

Simple Network Management Protocol (SNMP) enables enterprise-wide management systems to manage heterogeneous software and hardware environments from a single management console.

The following sections describe the contents and organization of this guide—*WebLogic SNMP Management Guide*.

- "Document Scope and Audience" on page 1-1

- "Guide to this Document" on page 1-2

- "Related Documentation" on page 1-2

- "New and Changed SNMP Features in This Release" on page 1-4

Document Scope and Audience

This document is a resource for systems administrators who use SNMP to monitor WebLogic Server.

The topics in this document describe the SNMP capabilities of WebLogic Server. The Administration Console Online Help provides specific, task-related information on configuring SNMP services in a WebLogic Server domain.

It is assumed that the reader is familiar with SNMP and general network management concepts. For background information on SNMP, refer to the documents listed in "Related Documentation" on page 1-2.

Guide to this Document

This document is organized as follows:

- This chapter, Introduction and Roadmap, describes the audience of the guide and provides pointers to related documentation.

- Chapter 2, "Understanding the WebLogic SNMP Agent and MIB," describes basic concepts of Simple Network Management Protocol as they apply to managing WebLogic Servers.

- Chapter 3, "Understanding WebLogic Server Trap Notifications," describes the characteristics of WebLogic enterprise-specific SNMP trap notifications.

- Chapter 4, "Understanding SNMP Proxies," describes how WebLogic Server can function as a master agent that proxies for other SNMP agents.

Related Documentation

For step-by-step instructions on configuring SNMP services in a WebLogic Server domain, see "Use SNMP to Monitor WebLogic Server" in the *Administration Console Online Help*.

For information on other technologies for monitoring WebLogic Server, see the following documents:

- *Developing Manageable Applications with JMX*

- *Configuring and Using the WebLogic Diagnostic Framework*

For background information on SNMP, see com.protocols.snmp SNMP FAQ Part 1 and Part 2.

Standards and Drafts

The SNMP protocol has been defined through a series of Requests for Comments (RFCs). The standards and drafts in Table 1-1 are available from www.faqs.org/rfcs.

Table 1-1 SNMP RFCs

RFC Number	Description
052	IAB Recommendations
1089	SNMP over Ethernet

RFC Number	Description
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)
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

New and Changed SNMP Features in This Release

This release contains no changes for SNMP features.

Changes to SNMP Features in the WebLogic Server 9.0 Release

WebLogic Server 9.0 significantly reorganized the WebLogic Server management data model, and as a result added new managed objects in the MIB and deprecated others. See Introduction and Roadmap in *WebLogic SNMP Management Guide* for WebLogic Server 9.0.

Understanding the WebLogic SNMP Agent and MIB

Simple Network Management Protocol (SNMP) enables enterprise-wide management systems to manage heterogeneous software and hardware environments from a single manager. Typically, the manager includes a user interface provided by a third-party SNMP management system. The WebLogic SNMP agent is a WebLogic Server® subsystem that gathers WebLogic management data (managed objects), converts it to SNMP communication modules (trap notifications), and forwards the trap notifications to third-party SNMP management systems. The WebLogic SNMP agent can also respond to direct requests from a manager using PDU format. The WebLogic SNMP agent supports the SNMPv1 and SNMPv2 protocols. All managed objects that WebLogic Server exposes for management through SNMP are defined in the WebLogic Server Management Information Base (MIB).

The following sections describe the SNMP management model and how WebLogic Server implements this model:

- "The SNMP Agent–Manager Model" on page 2-2

- "The Role of the SNMP Agent in a WebLogic Server Domain" on page 2-2

- "MIB for WebLogic Server" on page 2-4

- "Community Names for WebLogic Server" on page 2-7

For more information, refer to:

- "Use SNMP to monitor WebLogic Server" in the *Administration Console Online Help*

- "WebLogic SNMP Agent Command-Line Reference" in the *WebLogic Server Command Reference*

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

An agent gathers and sends data about managed resources in response to a request from a manager. Agents can also 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**.

To request and receive data from an agent, a manager relies on a database that describes the properties of managed resources and the services that the agent supports. This database is called a **Management Information Base** (MIB), and the properties that it describes are called **managed objects**. If an agent adds managed objects to its MIB, the manager must reload the corresponding MIB to discover the new objects.

The Role of the SNMP Agent in a WebLogic Server Domain

Each WebLogic Server domain contains a single SNMP agent that runs on the Administration Server. The agent provides a single point of contact for an SNMP manager to get information about managed objects on all server instances in the domain. See Figure 2-1. (For more information about domains, refer to "Understanding WebLogic Server Domains" in *Configuring and Managing WebLogic Server*.)

You can use the WebLogic SNMP agent to:

- Respond to simple GET requests from an SNMP manager (as shown in Figure 2-1) for the current value of WebLogic Server managed objects on any server instance in the domain.

  Note: WebLogic Server does not enable SNMP managers to set the values of managed objects or invoke management operations. SNMP managers can be used only to monitor WebLogic Server.

- Send trap notifications to SNMP managers when the Administration Server or any Managed Server starts or shuts down.

- Listen for specific log messages and send trap notifications to SNMP managers when WebLogic Server generates them.

- Listen for changes to the configuration of a WebLogic Server domain and send trap notifications when the change meets criteria you specify.

- Use JMX monitors to poll WebLogic Server resources periodically and send trap notifications to SNMP managers when the resources change in a way that you specify.

- Act as a proxy agent that passes requests from an SNMP manager to other (non-WebLogic) SNMP agents (such as an Oracle database agent) on the same machine.

To enable and configure the WebLogic SNMP agent, refer to "Configure the SNMP Agent" in the *Administration Console Online Help*.

Figure 2-1 SNMP Management of a WebLogic Domain
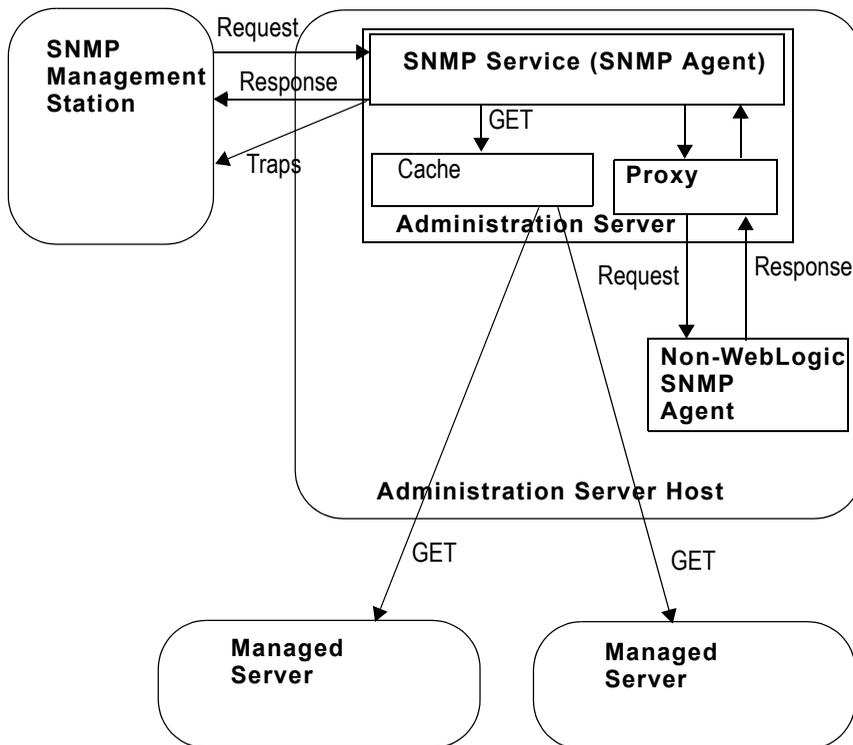
MIB for WebLogic Server

The BEA WebLogic SNMP MIB uses Abstract Syntax Notation.1 (ASN.1) to describe the resources that can be monitored through SNMP and the trap notifications that its SNMP agent can send to SNMP managers.

The WebLogic Server installer creates a copy of the MIB in the following location:
`BEA_HOME/weblogic90/server/lib/BEA-WEBLOGIC-MIB.asn1`

where `BEA_HOME` is the directory in which you install WebLogic Server. With each new release, WebLogic Server appends any new managed objects to the MIB. The object identifiers for existing managed objects do not change from one release to the next.

The following sections describe the WebLogic Server MIB:

- "Hierarchical Data Model" on page 2-4
- "Object Identifiers" on page 2-6
- "Browsing the MIB" on page 2-7

Hierarchical Data Model

WebLogic Server exposes thousands of data points in its management system. To organize this data, it provides a hierarchical data model that reflects the collection of services and resources that are available in a domain. For example, a WebLogic Server domain can contain multiple servers. Each server contains (or hosts) applications, and each application contains Web applications, EJBs, and other J2EE modules.

The WebLogic Server MIB reflects this hierarchy. For example, a WebLogic Server domain describes its overall configuration in a tabular managed object called `domainTable`. This tabular object refers to (contains) a collection of scalar objects, each of which describes some attribute of the domain. For example, `domainTable` contains a `domainServers` scalar object that names all servers in the domain. The `serverTable` object contains a `serverDeployments` scalar object, which describes all applications currently deployed on a server.

Tabular objects never directly contain object instances. Instead, tabular objects contain scalar objects, and the scalar objects contain object instances. For example, if you created two Managed Servers in a domain named `MS1` and `MS2`, the MIB contains one `serverTable` object, which in turn contains a `serverName` object. The `serverName` object contains two object instances that contain the value `MS1` and `MS2`. See Figure 2-2.

Figure 2-2 Hierarchy of Objects and Object Instances
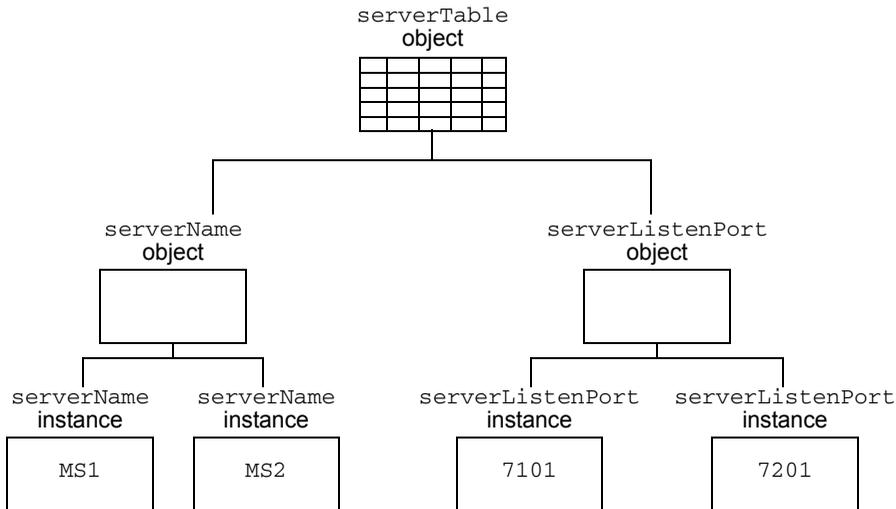
Configuration and Runtime Hierarchies

Instead of one large hierarchy for all of its management data, the WebLogic Server management data model consists of two hierarchies: one for its configuration data and another for the performance and monitoring data that are available only at runtime. All managed objects that describe runtime data contain the word "runtime" in their name; configuration managed objects do not. For example, the MIB contains a `domainTable` that describes a domain's configuration and a `domainRuntimeTable` that describes runtime data.

Relationship of the MIB to the WebLogic Server MBean Data Model

WebLogic Server provides hundreds of managed beans (MBeans) as part of its implementation of Java Management Extensions (JMX). JMX is a J2EE specification for programmatic access to a Web application server's management data, and an MBean is the representation of the management data and operations. JMX's purpose is the same as SNMP: provide standard communication of management information between agents and managers.

At the implementation level, the WebLogic Server SNMP agent and MIB form a protocol-specific layer on top of the WebLogic Server JMX implementation. If you are already familiar with the WebLogic Server JMX implementation, you will notice similarities in the data model for WebLogic Server MBeans and the organization of managed objects in the WebLogic Server MIB. However, there are some important differences:

- WebLogic Server enables JMX clients (similar to SNMP managers) to monitor a domain *and* to modify a domain configuration. WebLogic Server gives SNMP managers only read access to its management system.

- The data model for MBeans is a deep hierarchy, while the data model implied by the MIB is shallow. For example, a JMX client can navigate from a `DomainMBean` to its child `ServerMBeans`, and then to the children of each `ServerMBean`, and so on. The MIB, on the other hand, represents objects using unique identifiers. See "Object Identifiers" on page 2-6.

For more information about the WebLogic Server JMX implementation, see Understanding WebLogic Server MBeans in *Developing Manageable Applications with JMX*.

Object Identifiers

A MIB assigns a unique, immutable number called an **object identifier** (OID) to each managed object that it describes. Each OID consists of a left-to-right sequence of integers. This sequence defines the location of the object in the MIB tree and specifies a unique path through the tree to the object. Each node in the path 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, for example, BEA. 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 `.1.3.6.1.4.1.140.625`.

OIDs for Objects and Instances

The WebLogic Server MIB uses OIDs to reflect its hierarchical data model. For example, the OID for the `serverRuntimeTable` object is `.1.3.6.1.4.1.140.625.`**360**. The OID for the `serverRuntimeState` scalar object, which is contained by the `serverRuntimeTable` object is `.1.3.6.1.4.1.140.625.`**360.1.60**.

To identify a specific **instance** of an object, the WebLogic SNMP agent generates and appends an additional set of numbers to the object's OID. For example, the OID for an instance of `serverRuntimeState` would be `.1.3.6.1.4.1.140.625.`**360.1.60.32.102.100.48.98.101.102.100.99.102.52.98.97.48.48.49.102.57.53.51.50.100.102.53.55.97.101.52.56.99.99.97.99.**

The OID is persistent across instantiations of the object.

You can use the *WebLogic Server SNMP MIB Reference* to see the OIDs for managed objects, and the `snmpwalk` or `snmpgetnext` commands to see the OIDs for any managed object instance.

For more information, refer to "WebLogic SNMP Agent Command-Line Reference" in the *WebLogic Server Command Reference*.

Browsing the MIB

To browse the contents of the WebLogic Server MIB:

- Use a MIB browser. WebLogic Server does not provide a MIB browser, but most vendors of SNMP utilities do.

- Or, use a Web browser to view the *WebLogic Server MIB Reference* on the BEA e-docs Web site.

  Because the MIB Reference uses Javascript and DHTML to provide browsing capabilities that are similar to a MIB browser, you must use one of the following Web browsers:

  – Internet Explorer, version 5 or higher

  – Netscape Navigator, version 6 or higher

  – Opera 7 or higher

  – Mozilla

  – Phoenix

Community Names for WebLogic Server

To ensure that an SNMP manager 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 Administration Server (described in "Use SNMP to Monitor WebLogic Server" in the *Administration Console Online Help*), you specify the community name that the agent expects from the SNMP manager. If the agent receives an SNMP request with an incorrect community name, it generates an authenticationFailure trap and sends it to the source of the request.

Using Community Names to Specify Target Servers in Management Requests

Because a WebLogic Server domain can have multiple server instances concurrently active, a request from an SNMP manager that specifies only a managed object is potentially ambiguous. For example, the object serverUptime exists for each WebLogic Server instance in a domain.

To request a managed object on a specific Managed Server, when you send a request from an SNMP manager append the name of the server instance to the SNMP community name that it sends with the request as follows:

`community_prefix@server_name`

where `community_prefix` is the SNMP community name and `server_name` is the name of the targeted 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.

To request a managed object on the Administration Server, send a community name to the WebLogic SNMP agent with the following form:

`community_prefix`

To request a managed object for all server instances in a domain, send a community string with the following form:

`community_prefix@domain_name`

Understanding WebLogic Server 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. The WebLogic SNMP agent can generate traps that conform to the SNMPv1 or SNMPv2 protocols.

The following sections describe the trap notifications that the WebLogic SNMP agent can generate:

- "Format of WebLogic Trap Notifications" on page 3-1
- "Automatically Generated WebLogic SNMP Traps" on page 3-3
- "Log Message Traps" on page 3-4
- "Monitor Traps" on page 3-6
- "Attribute Change Traps" on page 3-9

To configure or delete WebLogic Server trap notifications, refer to "Use SNMP to monitor WebLogic Server" in the *Administration Console Online Help*.

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) that contains the fields shown in Figure 3-1.

Figure 3-1 SNMP Trap Packet

PDU type	enterprise	agent address	generic trap type	specific trap type	timestamp	variable bindings

The fields contain the following information:

- `PDU type`— Identifies the packet as a trap notification.

- `enterprise` — Contains `.1.3.6.1.4.140.625,` which indicates the trap is generated by the WebLogic SNMP agent.

- `agent address` — IP address of the Administration Server. (All traps are generated and sent from the Administration Server even if an event occurs on a Managed Server.)

- `generic trap type` — Contains an integer in the range of 0 to 6. Table 3-1 lists the values that the different types of WebLogic SNMP traps supply for the `generic trap type` field.

Table 3-1 Values for the Generic Trap Type Field

`generic trap type` Value	Generated When
0	The Administration Server starts. This is called a `coldStart` trap.
4	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 "Community Names for WebLogic Server" on page 2-7.) This is called an `authenticationFailure` trap.
6	All other WebLogic SNMP traps.

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` — Number that further qualifies an `enterpriseSpecific` trap. Table 3-2 lists the values that the different types of WebLogic SNMP traps supply for the `specific trap type` field.

Table 3-2 Values for the Specific Trap Type Field

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

- `timestamp` — 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. Later sections describe the name-value pairs for each type of trap notification:

  - "Automatically Generated WebLogic SNMP Traps" on page 3-3

  - "Variable Bindings in Log Message Traps" on page 3-5

  - "Variable Bindings in Monitor Traps" on page 3-8

  - "Variable Bindings in Attribute Change Traps" on page 3-9

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 3-3. Some of these traps include name–value pairs in the PDU to further describe the event.

Table 3-3 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 "Community Names for WebLogic Server" on page 2-7.)	none
serverStart	A 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:

- Message severity level

- Name of the subsystem that generated the message

- User ID under which the subsystem is running

- Unique message ID

- 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 SNMP Log Filters" in the *Administration Console Online Help*.

When the agent receives a message, it generates an SNMP log notification trap. (See Figure 3-2.)

Figure 3-2 Log Message Traps
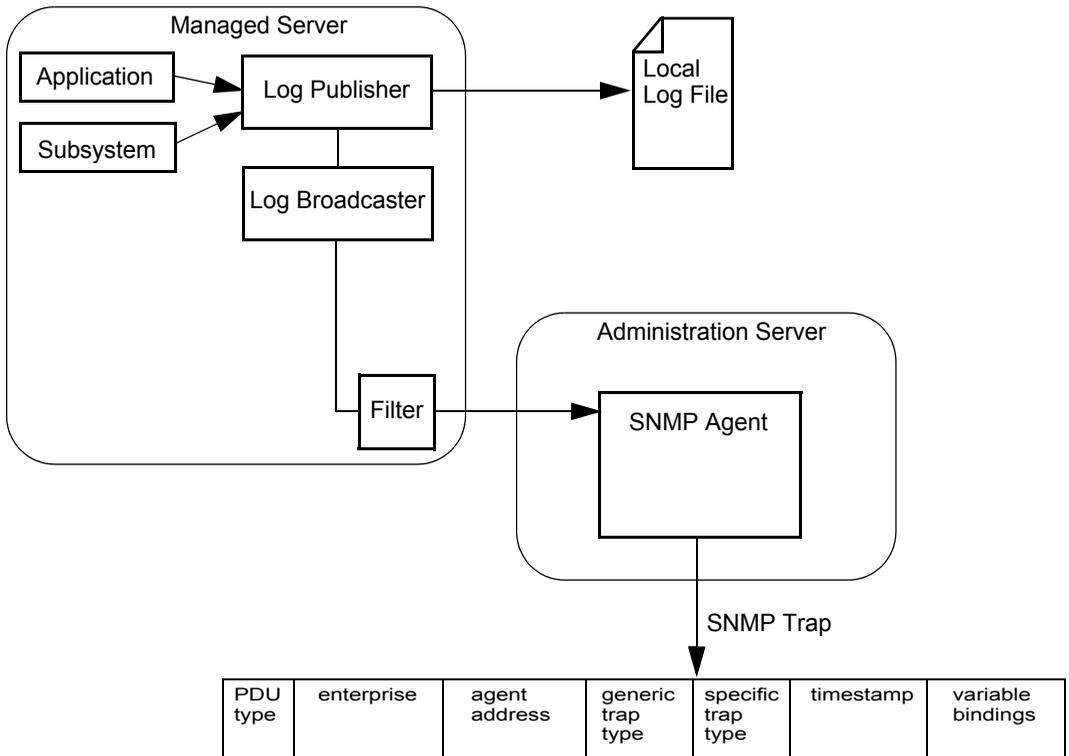
Variable Bindings in Log Message Traps

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

- trapTime — Time 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` — User ID from the log message. The user ID indicates the security context in which the log message was generated.

- `trapLogSubsystem` — Subsystem that generated the log message.

- `trapLogMsgId` — Log message ID from the log message.

- `trapLogSeverity` — Message severity level from the log message.

- `trapLogMessage` — Text of the log message.

For more information on log messages and the WebLogic Server logging subsystem, refer to "Understanding WebLogic Logging Services" in *Configuring Log Files and Filtering Log Messages*.

Monitor Traps

To periodically check the value of WebLogic resources for changes, you set up JMX monitors and configure the SNMP agent to listen for notifications from these monitors.

JMX is a J2EE specification for exposing management data; it is the foundation for the WebLogic Server management system. In the JMX specification, management data and operations are made public through managed beans (MBeans). The managed objects in the WebLogic Server MIB correspond to MBeans and MBean attributes. See "Relationship of the MIB to the WebLogic Server MBean Data Model" on page 2-5.

JMX monitors poll the WebLogic Server MBeans that are represented in the MIB 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 3-3.)

Figure 3-3   Monitor Traps
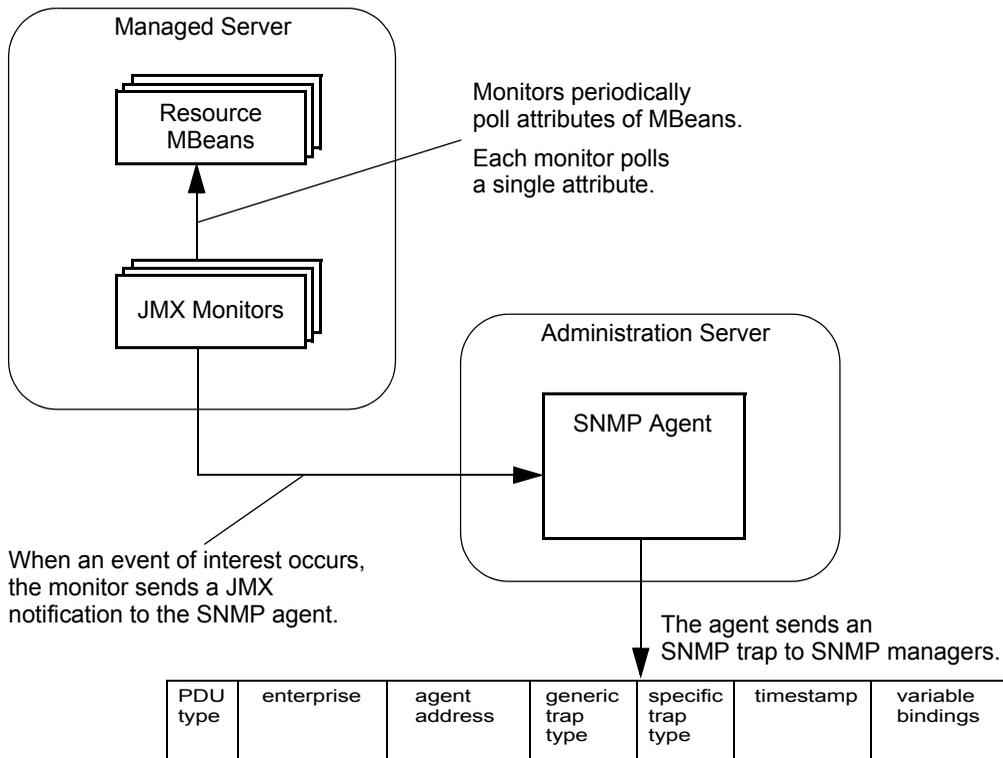
You can configure three types of JMX monitors, depending on the data type of the attribute that you want to observe (the *WebLogic Server MBean Reference* describes the type of data that an attribute returns):

- **Counter Monitor**

    A counter monitor observes MBean 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 counter monitors" 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 gauge monitors" 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 string monitors" 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` — Time at which the trap was generated.

- `trapServerName` — Local server whose attribute value generated the trap.

- `trapMonitorType` — Either `CounterMonitor`, `StringMonitor`, or `GaugeMonitor`.

- `trapMonitorThreshold` — ASCII representation of the threshold that triggered the trap.

- `trapMonitorValue` — ASCII representation of the value that triggered the trap.

- `trapMBeanName` — Name of the MBean that contained the attribute being monitored.

- `trapMBeanType` — Type of the MBean that contained the attribute being monitored.

- `trapAttributeName` — 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 JMX monitor to poll for changes in the current number of active JDBC connections. If the number of active connections 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 data source or the server's listen port has been changed.

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

Note: Creation of attribute changes for runtime MBeans is not supported. Only attributes of configuration MBeans support attribute change traps.

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.

Understanding SNMP Proxies

The following sections provide background information on WebLogic Server and SNMP proxy agents. For information on configuring WebLogic Server to be a proxy for other SNMP agents, refer to "Create SNMP Proxies" in the *Administration Console Online Help*.

- "SNMP Agent as Proxy for Other Agents" on page 4-1
- "The Microsoft Windows SNMP Service" on page 4-2

SNMP Agent as Proxy for Other Agents

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 enable 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.

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 (non-Weblogic) SNMP agents. Each of these will be a branch that encompasses the private MIB (or some part of that MIB) that the target agent is designed to manage.

Note: You cannot use the WebLogic SNMP agent as a proxy for SNMP agents in other WebLogic Server domains. For example, WebLogic domainA's SNMP agent cannot proxy requests to domainB's SNMP agent. This limitation is in effect because all WebLogic SNMP agents use the same MIB root.

Instead of proxying requests to multiple WebLogic Server domains, you can place all of your server instances in a single domain and send requests directly to each Managed Server. See "Using Community Names to Specify Target Servers in Management Requests" on page 2-7.

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. For information on configuring the WebLogic Server SNMP agent, see Configure the SNMP Agent in the *Administration Console Online Help*.

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.

Index

for WebLogic 3-2, 3-3

T

trap notification
    what it is 2-2
traps based on log messages 3-4

V

variable bindings
    in attribute change trap 3-9
    in log message trap 3-5
    in monitor trap 3-8, 3-9

W

WebLogic
    specific trap types 3-2, 3-3
WebLogic enterprise OID 3-2