Oracle® Enterprise Manager

SNMP Support Reference Guide

11*g* Release 1 (11.1.0.1.0)

E14091-02

September 2012

Oracle Enterprise Manager SNMP Support Reference Guide 11*g* Release 1 (11.1.0.1.0)

E14091-02

Contents

A Configuration Files

B Interpreting Variables of the Oracle Database MIB

List of Examples

List of Figures

List of Tables

Preface

This manual introduces the Oracle SNMP Support feature and describes the public and private MIBs that support its use with certain products. This information is provided to assist you in developing SNMP-based management applications for these products.

This preface contains the following sections:

Note that later versions of this and other Enterprise Manager books may be available on the Oracle Technology Network:

`http://www.oracle.com/technology/documentation/oem.html`

Audience

This manual is intended primarily for third-party developers of SNMP-based management applications. While these readers may have a good working knowledge of SNMP, they may not be as familiar with relational databases, or Oracle products in particular. A secondary audience for this manual includes database administrators, who may be familiar with Oracle products, but who may be less familiar with SNMP.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following books in the Enterprise Manager Grid Control documentation library:

- *Oracle Enterprise Manager Grid Control Basic Installation Guide*
- *Oracle Enterprise Manager Grid Control Advanced Installation and Configuration Guide*
- *Oracle Enterprise Manager Administration Guide*
- *Oracle Enterprise Manager Administrator's Guide for Software and Server Provisioning and Patching*

For the latest releases of these and other Oracle documentation, check the Oracle Technology Network at the following URL:

`http://www.oracle.com/technology/documentation/oem.html`

Grid Control also provides extensive online Help. Click **Help** at the top of any Grid Control page to display the online help window.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates code in examples.

Part I

List of Chapters

Part I covers chapters that provide an overview of Oracle SNMP support, procedures for configuring SNMP for Oracle, and other information pertaining to MIB variables.

Part I contains the following chapters:

1

Overview of Oracle SNMP Support

This chapter provides a brief overview of Oracle SNMP support, including:

The Simple Network Management Protocol (SNMP) is a protocol used for network management. SNMP enables a single application to first retrieve information, then push new information between a wide range of systems independent of the underlying hardware.

Designed primarily for database, network, and system administrators, Oracle SNMP support integrates the management of Oracle products into a number of existing, widely-used management systems. This feature enables key Oracle products running anywhere on an enterprise's network to be located, identified, and monitored by a management station running at a centrally located node, in much the same way and using some of the same tools as traditionally have been used to monitor the activity of the network itself. It thereby integrates the tasks of database and of network administrators, enabling both to use some of the same tools and to better integrate their tasks. Tools using SNMP traditionally provide powerful features for monitoring network components. Oracle extends this power to enable SNMP monitoring of some of its own products.

Benefits of Oracle SNMP Support

The primary benefits of Oracle SNMP support include the following:

- The monitoring of key Oracle products is quickly integrated into any management framework based upon SNMP.
- These Oracle products are located, identified, and monitored in real time across enterprise networks of any size.
- Administrators see standard Oracle icons that represent Oracle products in a network map. You can dynamically customize this map. In fact, administrators can define and customize various network maps for different purposes.

- Administrators see the current status of Oracle products, as shown by several status variables that are defined for each product in a management information base (MIB), or they can select which elements to view by their status.
- Administrators can anticipate exceptional conditions by defining thresholds and alerts, to respond to special situations as soon as they occur or to enable automatic responses.
- Administrators can more readily determine key characteristics of Oracle objects, such as database size, number of users, and activity level.
- Administrators can store and analyze historical data that has been obtained through SNMP.
- Providers of management applications can easily build customized solutions for Oracle customers because SNMP is an open standard.

Strictly speaking, Oracle SNMP support is intended more for monitoring Oracle products than for managing them. Oracle SNMP support is invaluable for tracking the status of an entire network of Oracle applications — first, to verify normal operations, and second, to spot and react to potential problems as soon as they are detected. However, for purposes of investigating and solving some problems, other Oracle tools such as Oracle SQL *Plus Worksheet may be more appropriate. This is because Oracle SNMP support is designed to query status, but not to change system parameters, whereas other tools are designed to set or tune system parameters.

Note: Oracle SNMP is not supported on HP OpenVMS platform.

How SNMP Works

SNMP is a standard internet protocol enabling certain nodes in a network, the management stations or managing nodes, to query other network components or applications for information concerning their status and activities. Such a query is known as an SNMP poll. The items that can be so polled are called managed elements. Managed elements can be network components such as bridges, routers, databases.

The software used by a management station is called a management framework or management platform. The management framework uses the SNMP protocol to request information from agents on the nodes being managed, and those agents send back the appropriate responses. The agents can also, independently of the framework, transmit messages called traps to well-known addresses in response to specific events. This is done to enable quick and possibly automatic reactions to the specific conditions that the traps indicate.

All requests sent to a given network node are handled by the same master agent. This agent redirects the requests to the appropriate managed elements on the node, in some cases using subagents. The protocol used for this is not yet standardized and is not SNMP. The information that SNMP can obtain is described in a structure called a Management Information Base (MIB), which is located on the node of the managed element.

Figure 1–1 shows the components of a management station and of a sample managed node.

The Components of SNMP

The components shown in Figure 1–1 are explained in the sections that follow.

Management Station

The management station refers to a node from which managed elements are monitored using the SNMP protocol. Typically, it is a standalone workstation that is on the same network as the managed elements. While this book will consistently use the term management station, other terms used for it include management console, management system, or managing node.

Requirements for Implementing Oracle SNMP Support

The following components are needed to implement Oracle SNMP support on the management station:

- A management platform or framework. At least one general purpose management application is required.
- A repository to store historical data regarding system activity, for reporting, displaying, or accounting. Some frameworks do offer an Oracle database as a repository, but in other cases the format of the repository is set.
- Various third-party specialized management applications. These are optional add-in components that often integrate with the framework and provide more in depth management of particular devices or applications.

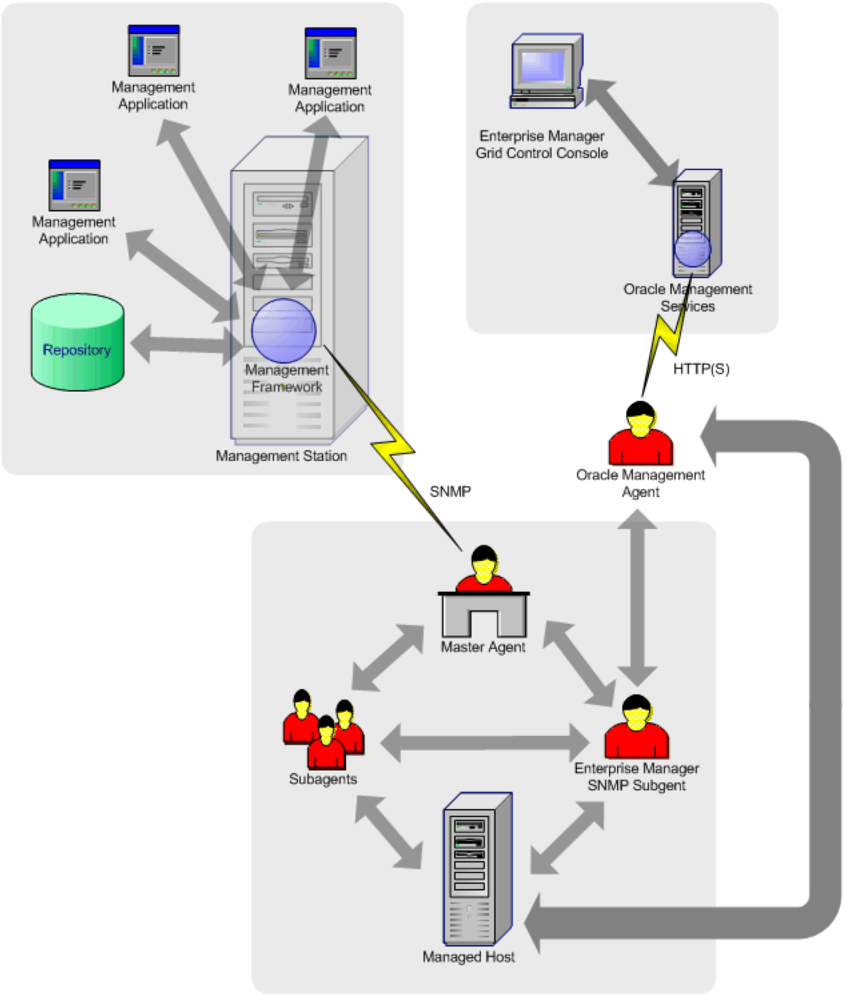
Note: These components are not part of Oracle SNMP support.

Management Framework

At the management station, the management framework uses SNMP to request management information from other nodes. The framework collects, graphs, and possibly acts on that SNMP data, and saves some or all of it in a repository for historical analysis and reporting. Management frameworks include many tools and options. In addition to directly requesting information from managed nodes, frameworks typically use daemons to alert them when a managed node has sent a trap in response to a specific set of conditions. The traps also can be used to trigger management applications.

Note: Oracle does not provide the management framework.

Figure 1–1 Basic Components of Oracle SNMP Support



Because most frameworks use SNMP as a basis for communication, Oracle products that support SNMP can be integrated into virtually every management framework. Examples of such frameworks include CA Unicenter, HP OpenView, Tivoli NetView, Aprisma Spectrum, Sun Solstice, Castle Rock SNMPc Network Manager.

Most of today's management frameworks also provide a selection of graphical objects that management applications may use to build a graphical user interface that serves their particular needs, such as:

- Maps illustrating logical or physical network topologies.
- Icons representing individual network components.
- Graphing tools such as dials, bar charts, line plots, and so on, for effective monitoring of management information variables.

Management Application

The management applications are the tools integrated with the management framework to accomplish more specialized network or database tasks. These applications contain virtually all of the sophisticated logic associated with network management.

A customized management application can work with one or more frameworks (on different management stations) or run independently. Because Oracle SNMP support is equally accessible to any type of provider, there are many different ways that applications can utilize it.

A fundamental management application, often shipped by default along with the management framework, is one that is capable of discovering the network topology and collecting some basic identification information about each discovered network entity or service. Such an application, for instance, may discover all hosts in a subnet along with their vendor, location, and status. Using this information, the management application can subsequently build up logical maps of the environment.

Managed Node

The managed node is a platform, such as a UNIX server, on which elements to be monitored reside.

Master Agent

The master agent is the process on a managed node that accepts queries, also called "polls", from the management framework and communicates with the elements to be managed in order to answer the query. It also can send SNMP Traps independently in response to specific conditions. Only one master agent can exist on each managed node. Any node that does not have a master agent will not be able to respond to SNMP requests, but this does not prevent other nodes on the network from doing so. In other words, it is not necessary that every node in a network be able to respond to SNMP, although this is normally desirable.

The master agent may be either monolithic or extensible. If it is monolithic, it communicates directly with the elements to be managed. Although such a management agent can manage multiple elements on the same node, the set of elements that it can manage is fixed when the management agent is created, because the monolithic agent itself is responsible for interfacing to the managed elements.

If, on the other hand, the master agent is extensible, it will use a specific subagent for each element it has to manage. That subagent is then responsible for interfacing to the element. In this scenario, new subagents can register with the master agent at any time, so new managed elements can be added dynamically.

Some operating systems supply only monolithic agents. In this case, Oracle provides a master agent that can effectively treat that monolithic master agent as a subagent, enabling new managed elements to be added to the node dynamically.

Subagent

The subagent is a process that receives queries for a particular managed element from the master agent and sends back the appropriate answers to the master agent. One subagent exists for each managed element residing on the managed node (with the exception that a single subagent can handle multiple Oracle database instances on the same node). The subagent(s) and master agent communicate using a multiplexing protocol dictated by the master agent. There is no standard protocol for this connection, and, while a few protocols are widely used, none is a designated standard.

How SNMP Communications Are Performed

SNMP is based on connectionless communication between the framework and the managed nodes. Because most management information does not demand reliable delivery, SNMP packets are transmitted from one node to a well-known address of another node, but no verification of successful delivery is made. The penalty for the light-weight, connectionless SNMP communication is paid by the management applications, which need to verify that SNMP transactions get completed successfully within a reasonable amount of time. If SNMP packets get lost in the network, the application cancels the associated transaction and possibly re-initiates it.

The most popular SNMP implementation uses the User Datagram Protocol (UDP) over the Internet Protocol (IP), although implementations also exist over other protocols, such as Novell's Internetwork Packet Exchange (IPX) and Apple's AppleTalk.

Oracle Enterprise Manager and SNMP

Enterprise Manager web-based architecture provides a monitoring and administrative framework that is globally scalable and easy to deploy. SNMP extends Enterprise Manager's monitoring capabilities to any SNMP-enabled entity within your managed environment. For more information about Enterprise Manager, its functionality, and architecture, see the *Oracle Enterprise Manager Concepts* guide.

SNMP Trap Notification

Using the Enterprise Manager notification system, you can take advantage of SNMP-enabled third-party applications such as HP OpenView. For example, you may want to send third-party applications a notification from Enterprise Manager when a certain metric has exceeded a threshold. Using SNMP Traps with the notification system is a simple matter of defining a notification method that uses an SNMP Trap. Instructions for setting up a notification method using an SNMP Trap can be found in the next chapter.

SNMP Fetchlets

Fetchlets are parameterized data access mechanisms available to map relevant data from a managed element into Enterprise Manager's metric format. In the standards area, Enterprise Manager currently uses SNMP Fetchlets to fetch information from SNMP-enabled entities within your managed environment.

The SNMP Fetchlet mechanism is defined in the "query descriptor" section of the **metadata.xml** configuration file. Example 1–1 shows a sample metadata.xml configuration file that has SNMP Fetchlets defined for collecting metric information.

Example 1–1   SNMP Fetchlets Used in metadata.xml Configuration File

```
<Metric NAME="Load" TYPE="TABLE">
    <Display>
      <Label NLSID="cisco_router_load">Load</Label>
    </Display>

    <TableDescriptor>
     <ColumnDescriptor NAME="busyPer" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
          <Label NLSID="cisco_router_load_busyPer">CPU Busy in the last 5 seconds
(%)</Label>
        </Display>
     </ColumnDescriptor>

     <ColumnDescriptor NAME="avgBusy1" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
```

```
          <Label NLSID="cisco_router_load_avgBusy1">CPU Busy in the last
minute(%)</Label>
        </Display>
     </ColumnDescriptor>

     <ColumnDescriptor NAME="avgBusy5" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
          <Label NLSID="cisco_router_load_avgBusy5">CPU Busy in last 5 minutes
(%)</Label>
        </Display>
     </ColumnDescriptor>
    </TableDescriptor>

    <QueryDescriptor FETCHLET_ID="Snmp">
      <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
      <Property NAME="hostname" SCOPE="INSTANCE">agentHost</Property>
      <Property NAME="TABLE"    SCOPE="GLOBAL">FALSE</Property>
      <Property NAME="OIDS"     SCOPE="GLOBAL">1.3.6.1.4.1.9.2.1.56.0
                                               1.3.6.1.4.1.9.2.1.57.0
                                               1.3.6.1.4.1.9.2.1.58.0
</Property>
    </QueryDescriptor>
</Metric>
```

For Enterprise Manager 10g Grid Control Release 2 or lower, that is 10.2.0.2 or 10.2.0.1, you can only collect integer and string values from third-party, SNMP agents. This is because only SNMP version v1 is supported in the Fetchlet. Therefore, any integer value collected from the third-party, SNMP agent through SNMP v1 is accommodated in 32 bits.

For Enterprise Manager 10g Grid Control Release 3 or higher, that is 10.2.0.3 or higher, you can use *counter64* data type objects in your metadata.xml file to collect not only 32-bit integer and string values, but also 64-bit integer values.

By this, we can classify SNMP agents into two broad categories, mainly:

- SNMP v1 Agent that responds only to SNMP v1 queries
- SNMP v2 Agent that responds to SNMP v1 and v2 queries

In order to collect these counter64 values, you need to add one more query descriptor property called VERSION in the metadata.xml file. This is to intimate the Fetchlet to make appropriate version of SNMP request to fetch the data from the SNMP target.

For example, if the Fetchlet has to collect *ifHCInOctets* (64-bit value) attribute from the third-party, SNMP target, then the query descriptor definition in the metadata.xml file will look like this:

```
<QueryDescriptor FETCHLET_ID="Snmp">
      <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
      <Property NAME="hostname" SCOPE="INSTANCE">NAME</Property>
      <Property NAME="COMMUNITY" SCOPE="INSTANCE"
      OPTIONAL="TRUE">CommunityString</Property>
      <Property NAME="VERSION" SCOPE="GLOBAL">V2c</Property>
      <Property NAME="OIDS" SCOPE="GLOBAL">1.3.6.1.2.1.31.1.1.1.6 </Property>
</QueryDescriptor>
```

If the VERSION property value is set to V1 or if the VERSION property is not at all mentioned in the query descriptor, then the Fetchlet make only SNMP v1 query to collect the metric values, that is it collects only 32-bit integer and string values.

There may be occasions when you will want the Fetchlet to collect some metric columns from SNMP v1 Agent and some from SNMP v2 Agent. In this case, define the metrics using *ValidIf* tag based on the SNMP target version, that is if SNMP version is != v1, then counter64 data type property needs to be included in the metric definition; otherwise this property should be ignored. Here, the SNMP target version can be identified using a dynamic property.

For example, if you want to collect 1.3.6.1.2.1.31.1.1.1.5 (integer type) from SNMP v1 Agent switch device, and 1.3.6.1.2.1.31.1.1.1.5 (integer type) and 1.3.6.1.2.1.31.1.1.1.6 (Counter64 type) from SNMP v2 Agent switch, then your metric definition should look like this:

```
<Metric NAME="Table"  TYPE="TABLE">
   <ValidIf>
   <CategoryProp NAME="SnmpAgentVerision" CHOICES="V1"/>
   </ValidIf>

   <Display>
   <Label NLSID="SwithIfTable">SwitchInterface</Label>
   </Display>

   <TableDescriptor>
   <ColumnDescriptor NAME="Column1" TYPE="NUMBER">
      <Display>
      <Label NLSID="Column1">Column1 Value</Label>
      </Display>
   </ColumnDescriptor>
   </TableDescriptor>

   <QueryDescriptor FETCHLET_ID="Snmp">
   <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
   <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
   <Property NAME="COMMUNITY" SCOPE="INSTANCE"
   OPTIONAL="TRUE">CommunityString</Property>
   <Property NAME="TABLE" SCOPE="GLOBAL">TRUE</Property>
   <Property NAME="VERSION" SCOPE="GLOBAL">V1</Property>
   <Property NAME="OIDS" SCOPE="GLOBAL"> 1.3.6.1.2.1.31.1.1.1.5 </Property>
   </QueryDescriptor>
</Metric>


<Metric NAME="Table"  TYPE="TABLE">
   <ValidIf>
   <CategoryProp NAME="SnmpAgentVerision" CHOICES="V2"/>
   </ValidIf>

   <Display>
   <Label NLSID="SwithIfTable">SwitchInterface</Label>
   </Display>

   <TableDescriptor>
   <ColumnDescriptor NAME="Column1" TYPE="NUMBER">
      <Display>

      <Label NLSID="Column1">Column1 Value</Label>
      </Display>
   </ColumnDescriptor>

   <ColumnDescriptor NAME="Column2" TYPE="NUMBER">
      <Display>
      <Label NLSID="Column2">Column2 Value</Label>
      </Display>
   </ColumnDescriptor>
   </TableDescriptor>

   <QueryDescriptor FETCHLET_ID="Snmp">
   <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
   <Property NAME="hostname" SCOPE="INSTANCE">NAME</Property>
   <Property NAME="COMMUNITY" SCOPE="INSTANCE"
   OPTIONAL="TRUE">CommunityString</Property>
   <Property NAME="TABLE" SCOPE="GLOBAL">TRUE</Property>
   <Property NAME="VERSION" SCOPE="GLOBAL">V2c</Property>
   <Property NAME="OIDS" SCOPE="GLOBAL"> 1.3.6.1.2.1.31.1.1.1.5
                                      1.3.6.1.2.1.31.1.1.1.6
   </Property>
   </QueryDescriptor>
</Metric>
```

Here, the *SnmpAgentVersion* is being calculated by using the following dynamic property definition:

```
<DynamicProperties NAME="SnmpVersionConfig" FORMAT="ROW" OPT_PROP_
LIST="SnmpVersionIdentifier>
   <QueryDescriptor FETCHLET_ID="Snmp">
   <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
   <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
   <Property NAME="COMMUNITY" SCOPE="INSTANCE"
   OPTIONAL="TRUE">CommunityString</Property>
   <Property NAME="TIMEOUT" SCOPE="INSTANCE" OPTIONAL="TRUE">Timeout</Property>
   <Property NAME="VERSION" SCOPE="GLOBAL">V1</Property>
   <Property NAME="OIDS" SCOPE="GLOBAL"></Property>
   1.3.6.1.6.3.1.1.6.1.0
   </QueryDescriptor>
</DynamicProperties>

<DynamicProperties NAME="SnmpVersionValues" FORMAT="ROW" PROP_
LIST="SnmpAgentVersion">
   <QueryDescriptor FETCHLET_ID="VersionRangeComputer">
   <Property NAME="Version" SCOPE="INSTANCE"
   OPTIONAL="TRUE">SnmpVersionIdentifier</Property>
   <Property NAME="DefaultRange" SCOPE="GLOBAL">V1</Property>
   <Property NAME="V2" SCOPE="GLOBAL">0;</Property>
   </QueryDescriptor>
</DynamicProperties>
```

First, *SnmpVersionIdentifier* is being calculated based on one of the SNMP v2 MIB standard OID 1.3.6.1.6.3.1.1.6.1.0. Any bi-lingual standard Agent must have implemented this OID.

So when we try to collect the value for this OID via V1 request, if the agent is an SNMP v2 Agent, then we should get an integer value (range from 0-2147483647). However, if the agent is an SNMP v1 Agent, then we will get *nosuchname* error for this request.

Finally, an understandable format of *SnmpAgentVersion* values (v1 or v2) is being calculated using the *VersionRangeComputer* Fetchlet based on the value of *SnmpVersionIdentifier*.

Note: In this example, you can also specify any other application specific OID (instead of 1.3.6.1.6.3.1.1.6.1.0) to discover the *SnmpAgentVersion*, provided you know that that particular OID is implemented in the target SNMP agent and can be used to differentiate v1 and v2 SNMP Agents.

Note: For more information about Fetchlets, refer to Oracle Enterprise Manager Extensibility Guide available at:

`http://www.oracle.com/technology/documentation/oem.html`

SNMP Receivelets

While monitoring third-party entities in your managed environment, if the status of a third-party network element turns unavailable or if its metric severity conditions (metric thresholds) are met or exceeded, the SNMP Agent of that third-party network element sends a notification to Oracle Management Agent. These notifications may be in the form of SNMP Traps that get triggered asynchronously upon reaching the performance thresholds, and without any requests from Oracle Management Agent.

Since these traps are based on SNMP, Oracle Management Agent uses SNMP Receivelets to receive and translate these SNMP Traps into a form compatible with Oracle Management Service.

Once the SNMP Traps are received, the SNMP Receivelet extracts information pertaining to only those object identifiers (OIDs) that are defined in the “push descriptor“ section of the metadata.xml configuration file (that is only for those third-party network elements that need to be monitored by Enterprise Manager). Example 1–2 shows how SNMP Receivelets are defined in the metadata.xml configuration file.

Example 1–2 SNMP Receivelet Defined in metadata.xml Configuration File

```
<Metric NAME="Load" TYPE="TABLE">
    <Display>
      <Label NLSID="cisco_router_load">Load</Label>
    </Display>

    <TableDescriptor>
     <ColumnDescriptor NAME="busyPer" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
          <Label NLSID="cisco_router_load_busyPer">CPU Busy in the last 5 seconds
(%)</Label>
        </Display>
     </ColumnDescriptor>

     <ColumnDescriptor NAME="avgBusy1" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
          <Label NLSID="cisco_router_load_avgBusy1">CPU Busy in the last
minute(%)</Label>
        </Display>
     </ColumnDescriptor>

     <ColumnDescriptor NAME="avgBusy5" TYPE="NUMBER" IS_KEY="FALSE">
        <Display>
```

```
          <Label NLSID="cisco_router_load_avgBusy5">CPU Busy in last 5 minutes
(%)</Label>
        </Display>
     </ColumnDescriptor>
    </TableDescriptor>

<PushDescriptor RECVLET_ID="SNMPTrap">
<Property NAME="MatchEnterprise" SCOPE="GLOBAL">1.3.6.1.4.1.529</Property>
  <Property NAME="MatchGenericTrap" SCOPE="GLOBAL">6</Property>
  <Property NAME="MatchSpecificTrap" SCOPE="GLOBAL">50</Property>
  <Property NAME="MatchAgentAddr" SCOPE="INSTANCE">AdminAddress</Property>
  <Property NAME="EventbusyPerOID"
SCOPE="GLOBAL">1.3.6.1.4.1.9.2.1.56.0</Property>
  <Property NAME="SeverityCode" SCOPE="GLOBAL">CRITICAL</Property>
 </PushDescriptor>
</Metric>
```

Whenever an SNMP Trap is sent by the SNMP agent upon reaching the threshold value, the SNMP Receivelet receives those traps based on the SNMP Agent configuration, translates them to an Enterprise Manager understandable format (like event or datapoint based on the push descriptor information), and stores that information (in XML files) in the upload directory. The Upload Manager checks for such new files in the upload directory, and then uploads those files onto Oracle Management Service. Enterprise Manager, then, accesses the Oracle Management Service to extract the collected information and display it to the user.

Note: For more information about Receivelets, refer to Oracle Enterprise Manager Extensibility Guide available at:

`http://www.oracle.com/technology/documentation/oem.html`

Enterprise Manager SNMP Subagent

In Enterprise Manager 9.x and earlier releases, the Enterprise Manager SNMP Subagent functionality was integrated with Oracle Management Agent (formerly known as the Intelligent Agent). For Enterprise Manager release 10.x, SNMP subagent functionality now exists as a standalone executable that is installed along with Oracle Management Agent. The Enterprise Manager SNMP Subagent listens to SNMP requests and queries the database to obtain results from the subagent.

Management Applications can directly communicate with the Enterprise Manager SNMP Subagent using SNMP protocol on supported platforms. The Enterprise Manager SNMP Subagent provides access to Oracle's database Management Information Base (MIB) variables.

Oracle SNMP support is provided for the following:

- Oracle Database Server
- Oracle Network Listener Server
- Oracle Enterprise Manager

Since Oracle Management Agent and Enterprise Manager SNMP Subagent are separate processes, the Enterprise Manager SNMP Subagent cannot send out SNMP Traps. Hence, you cannot tie an alert to with the sending of an SNMP Trap, as was possible with 9.x and earlier versions of the Intelligent Agent. However, the Enterprise

Manager SNMP Subagent does send standard RDBMS traps when the database goes down.

Management Information Base (MIB)

While SNMP allows Enterprise Manager to send information to third-party SNMP-enabled applications, there may be situations where you want SNMP-enabled applications to obtain information from Enterprise Manager. This is accomplished with the help of MIB variables, and by signing up for SNMP Traps.

IMPORTANT: A valid Diagnostic Pack license is required to use the Enterprise Manager MIB variables.

A MIB is a text file, written in ASN.1 notation, which describes the variables containing the information that SNMP can access. The variables described in a MIB, which are also called MIB objects, are the items that can be monitored using SNMP. There is one MIB for each element being monitored. Each monolithic or subagent consults its respective MIB in order to learn the variables it can retrieve and their characteristics. The encapsulation of this information in the MIB is what enables master agents to register new subagents dynamically — everything the master agent needs to know about the subagent is contained in its MIB. The management framework and management applications also consult these MIBs for the same purpose. MIBs can be either standard (also called public) or proprietary (also called private or vendor).

The actual values of the variables are not part of the MIB, but are retrieved through a platform-dependent process called “instrumentation”. The concept of the MIB is very important because all SNMP communications refer to one or more MIB objects. What is transmitted to the framework is, essentially, MIB variables and their current values.

All MIBs are, in fact, part of one large hierarchical structure, with leaf nodes containing unique identifiers, data types, and access rights for each variable and the paths providing classifications. There is a standard path structure that includes branches for private subtrees. A portion of this structure is shown in Figure 1–2.

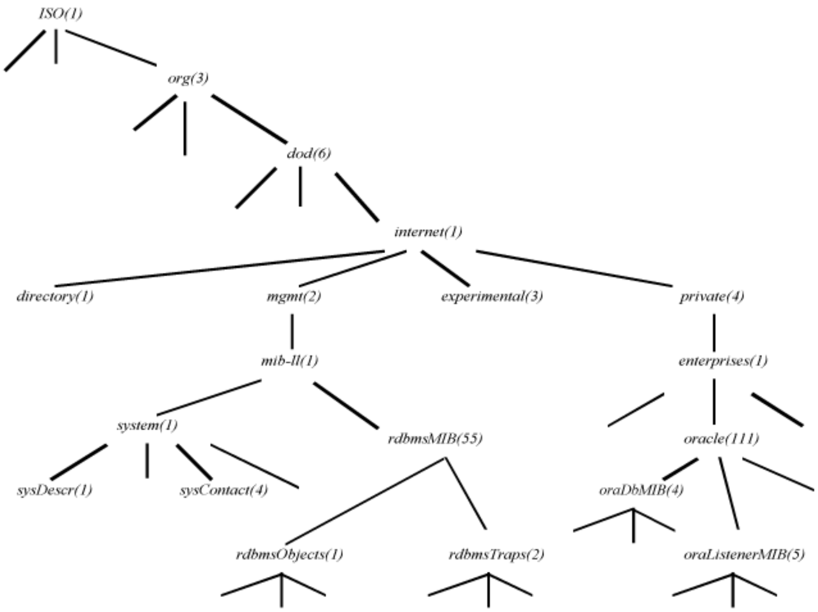
- the variable’s name
- the Object Identifier (OID) of the variable
- the variable’s data type
- the access rights associated with the variable
- a textual description of the meaning of the variable

The variable’s name is intended to be descriptive, whereas the OID is a number that describes the path taken through the tree to reach that variable. For example, the variable named sysContact, identified by the OID 1.3.6.1.2.1.1.4 (meaning iso.org.dod and so on), is a read/write string variable that contains contact information about the administrator of the underlying system.

All objects contained under the mgmt branch of Figure 1–2 (in other words, all objects with OIDs beginning 1.3.6.1.2) are considered standard and are tightly regulated by the Internet Engineering Task Force (IETF). For example, the standard RDBMS MIB lives under the mgmt branch and is supported by all relational database servers that claim to be SNMP-enabled. Oracle further adds its own MIB objects under the private branch to increase the manageability of its products. The following MIBs are specific to Oracle Services and are found under the {private.enterprise.oracle} branch:

- the Oracle Database MIB
- the Oracle Listener MIB

Figure 1–2 The MIB Hierarchy



Each leaf of this tree provides the following information about one MIB variable:

For more information on MIB, see Chapter 3, "Oracle MIB Overview" and Chapter 4, "Reading the MIB Variable Descriptions".

2

Configuring SNMP for Oracle

This chapter covers procedures for:

- Configuring SNMP on Microsoft Windows Platforms
- Configuring SNMP on UNIX Platform
- Configuring SNMP with Enterprise Manager

Keep in mind that the SNMP configuration is operating system specific. For more comprehensive configuration information, see the installation or configuration guide specific to your platform.

> **Note:** The SNMP configuration is case sensitive. Specify all occurrences of community names in the same case.

Configuring SNMP on Microsoft Windows Platforms

You can configure SNMP on Microsoft Windows XP, Windows 2000, Windows 2003 Server, or Windows NT platform. The configuration procedures given in this section are based on Windows XP platform. However, the same procedures apply to Windows 2000, Windows 2003, and Windows NT platforms.

Before you integrate SNMP with Enterprise Manager SNMP Subagent, complete the following tasks:

- Installing the SNMP Network Service
- Configuring the SNMP Network Service

After you integrate SNMP with Enterprise Manager SNMP Subagent, complete the following tasks:

- Starting the SNMP Environment
- Testing the SNMP Environment

Installing the SNMP Network Service

To install the SNMP network service on Windows XP:

1. From the **Start** menu, select **Settings**, then **Control Panel**, and then **Add or Remove Programs**.
2. On the Add or Remove Programs window, select the **Add/Remove Windows Components** option.

   The Windows Components Wizard lists all the components of Windows XP.

3. From the list of components displayed on the Windows Components Wizard, select the **Management and Monitoring Tools** component, and then click **Next**.
4. On the Files Needed dialog, provide the location of the files that are needed for completing the installation, and then click **OK**.

> **Note:** The files needed for completing the installation vary from one platform to another. For example, it can be Immib2.dll file and snmpmib.dll file for Windows 2000 platform, msi.mfl file for Windows 2003 Server platform, and so on.

5. After the installation completes, check to make sure the installation was successful by selecting **Settings**, then **Control Panel**, then **Administrative Tools**, and then **Services**.

   The Services window lists all the services installed on your computer. Under the Name column check for SNMP Service and SNMP Trap Service.

Configuring the SNMP Network Service

Follow these steps to configure the SNMP Network Service on Windows XP:

1. From the **Start** menu, select **Settings**, then **Control Panel**, then **Administrative Tools**, and then **Services**.
2. Select **SNMP Service** from the list of services; right-click and select **Properties**.
3. On the Agent tab, complete the following:
   - **Contact:** Contact information for this computer.
   - **Location:** Physical location of this computer.
   - **Service:** Select all Service Types (Physical, Applications, Datalink and subnetwork, Internet, and End-to-end).
4. On the Traps tab, add the following:
   - **Community Name:** The community is usually public.
   - **Trap Destinations:** Specify either the host name or IP addresses. This should be for all machines working with SNMP aware products that require traps. If you specify a host name, the system must be able to locate the corresponding IP address.

Integrating SNMP with Enterprise Manager SNMP Subagent

Enterprise Manager SNMP Subagent supports SNMP, allowing third-party system management frameworks to use SNMP to receive SNMP Traps directly from Oracle Management Agent. By configuring Oracle Management Agent to recognize SNMP requests from the master agent, third-party systems can gather relevant data.

Installing the Peer Executables

Installing the peer executables is the first step toward integrating SNMP with Enterprise Manager SNMP Subagent. Oracle SNMP Peer Master Agent (AGNTSVC.EXE) and Oracle SNMP Peer Encapsulator (ENCSVC.EXE) are the two peer executables that are bundled with Enterprise Manager SNMP Subagent. When you install Enterprise Manager SNMP Subagent, these peer executables also get installed.

1. After the installation is complete, select **Settings**, then **Control Panel**, then **Administrative Tools**, and then **Services** to see the following new services created:
   - Oracle SNMP Peer Master Agent
   - Oracle SNMP Peer Encapsulator

In the next section, you will learn how to configure the SERVICES file to have these peer executables listen on an unused port.

Configuring the SNMP Services File

In order for the SNMP Master Agent to communicate with both the standard SNMP services and the Enterprise Manager SNMP Subagent, the SNMP services file must be configured properly.

1. Specify an unused port where the Oracle SNMP Peer Encapsulator and Microsoft SNMP Service should be listening. Microsoft SNMP Service typically uses port 1161.

   The port is specified in the SERVICES file located in the following directory:

   ```
   C:\WINDOWS\system32\drivers\etc
   ```

   > **Note:** In Windows 2000 and Windows NT, the SERVICES file is located in the C:\WINNT\system32\drivers\etc directory.

   Make sure that you have the following lines in the file:

   ```
   snmp        1161/udp               #SNMP
   snmptrap    1162/udp   snmp-trap   #SNMP trap
   ```

   > **Note:** If an entry for SNMP already exists in the file, change the port from 161 (default number) to another available port (1161 in this example).

2. In the same location, check that the HOSTS and LMHOSTS.SAM files contain the mappings of IP addresses to host names for all computers in the SNMP setup.

   System performance will improve if more computer addresses can be resolved locally. Even if you use DHCP and WINS, adding the IP addresses will speed up the SNMP integration.

Modifying the Oracle SNMP Peer Master Agent Configuration File

Use the following procedure to modify the Oracle SNMP Peer Master Agent configuration file to reflect the SNMP configuration.

1. Specify the port where the SNMP Master Agent is listening.

   The port is specified in the TRANSPORT section of the MASTER.CFG file located in the ORACLE_HOME\NETWORK\ADMIN directory.

   For example, add the following section to the file:

   ```
   TRANSPORT   ordinary     SNMP
   OVER UDP SOCKET
   AT PORT 161
   ```

> **Note:** It is recommended that you use port 161 for the Oracle SNMP Peer Master Agent since port 161 is reserved for the SNMP framework. However, you may specify a different available port if you can configure the management application on your NMS to send SNMP requests to the master agent listening on this port.

2. Specify the authentication in the COMMUNITY section of the MASTER.CFG file.

```
COMMUNITY public
ALLOW ALL OPERATIONS
USE NO ENCRYPTION
```

3. If you need to send traps, specify the IP address (10.10.10.10 in this example) of that computer in the MANAGER section of the MASTER.CFG file.

```
10.10.10.10
SEND ALL TRAPS
WITH COMMUNITY public
```

   You only need to modify the manager section if you need to send traps, or use software that can manipulate and react to SNMP Traps, such as HP Openview. For performance reasons, specify IP addresses here, instead of host names.

4. If you have more than one Oracle SNMP Peer Master Agent running on a single computer, edit the Oracle SNMP Peer Encapsulator configuration file, ENCAPS.CFG. This file is located in the ORACLE_HOME\NETWORK\ADMIN directory and is used to specify which non-PEER master agents are to be encapsulated.

   You must add at least an AGENT entry, including MIB-subtrees manageable by NMS, for the encapsulated master agent.

   See the following example:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
         1.3.6.1.2.1.2,
         1.3.6.1.2.1.3,
         1.3.6.1.2.1.4,
         1.3.6.1.2.1.5,
         1.3.6.1.2.1.6,
         1.3.6.1.2.1.7,
         1.3.6.1.2.1.8,
         1.3.6.1.4.1.77
FORWARD ALL TRAPS;
```

> **Note:** The port (1161 in this example) must match the one you specified in the SERVICES file. Also, the community name must match previous definitions.

Starting the SNMP Environment

After you complete the configuration, the next step is to start the SNMP services. It is important that you:

- Verify UDP port 161 (or 1161) is unused
- Start and stop the services in the correct order

Verifying Active UDP Ports

Before you can start the SNMP environment, you need to verify that no other services are running on the designated UDP ports.

1. From the ORACLE_HOME\BIN directory, type the following command:

   ```
   netstat -p udp -a
   ```

   The results of this command list all active UDP ports on the computer. Port 161, and the SNMP port you specified in the SERVICES file must be unused.

Manually Starting SNMP Services

1. From the **Start** menu, select **Settings**, **Control Panel**, **Administrative Tools**, and then **Services**. In the Services window, check the status of Oracle SNMP Peer Master Agent, Oracle SNMP Peer Encapsulator, and SNMP Service. If they were previously started, terminate these programs now.
2. Check the status of Enterprise Manager SNMP Subagent. If it was previously started, stop it now.
3. Start the SNMP services in the following order:
   - SNMP Service
   - Oracle SNMP Peer Master Agent
   - Oracle SNMP Peer Encapsulator
   - Enterprise Manager SNMP Subagent

     Tip: To start, stop, or request the status of Enterprise Manager SNMP Subagent, use the Enterprise Manager command line utility (emctl). See Controlling Enterprise Manager SNMP Subagent for instructions.

     Note: The Oracle SNMP Peer Encapsulator is required only when you have more than one Enterprise Manager SNMP Subagent installed and configured on the computer.

Starting and Stopping SNMP Services by Batch File

You can also launch all relevant SNMP services by creating a batch file to automate the process. For example:

```
NET START "SNMP"
NET START "SNMP Trap Service"
Start C:\ORACLE_HOME\BIN\emctl start agent C:\ORACLE_HOME\NETWORK\ADMIN\master.cfg
C:\ORACLE_HOME\NETWORK\LOG\master.log
NET START "Service Name"
```

Testing the SNMP Environment

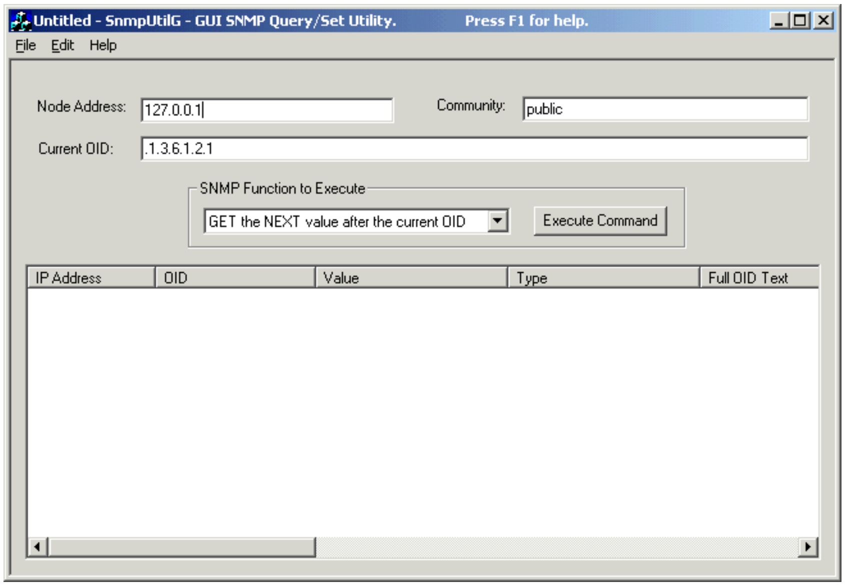
Once you have successfully installed SNMP, you are ready to query the MIBs from your Network Management Station.

One way to test the SNMP environment is by using the SNMPUTILG tool. The Windows XP Service Pack 2 Support Tools contains this SNMP troubleshooting tool that you can use to get basic information about the SNMP setup.

Note: You must download the Windows XP Service Pack 2 Support Tools separately at http://www.microsoft.com. Also, for other platforms like Windows 2003 Server, Windows 2000, and Windows NT, look for the SNMPUTILG.EXE tool in their respective service packs or resource kits.

The GUI SNMP Query/Set Utility (SNMPUTILG.EXE) (Figure 2–1) replaces the previous command line SNMPUTIL.EXE tool.

Figure 2–1 GUI SNMP Query Set/Utility



Untitled - SnmpUtilG - GUI SNMP Query/Set Utility. Press F1 for help.

Node Address: 127.0.0.1

Community: public

Current OID: .1.3.6.1.2.1

SNMP Function to Execute: GET the NEXT value after the current OID

Testing the Value of an OID

To test the value of an OID:

1. Open the GUI SNMP Query/Set Utility window (Figure 2–1) and change the following fields to reflect your values.

   - **Node Address:** IP address of your machine. This computer must be running the SNMP service.
   - **Community Name:** The community is usually `public`.
   - **Current OID:** .1.3.6.1.2.1.1.1.0
   - **SNMP Function to Execute:** Select **GET the value of the current OID**.

2. Click **Execute Command**.

   If the master agent is running correctly, it will return the sysDescr system identifier.

Testing Oracle Specific Values

Issue the following OID to test for Oracle specific values:

1. Follow the instructions in the Testing the Value of an OID section.
2. For the Current OID value, use .1.3.6.1.4.1.111.4.1.6.1.1. “index“

Where index is the value specified in the SNMP_RW.ORA file for the database you wish to query.

Configuring SNMP on UNIX Platform

The following configuration instructions are for UNIX platforms using O/S native or master agents that use the Peer Technology.

Configuring SNMP on UNIX platforms requires the following:

- Using only an SNMP Console
- Configuring the SNMP Master Agent
- Starting the SNMP Agents
- Loading the Oracle MIBs

> **Note:** For IBM AIX systems, see "Configuring SNMP on IBM AIX Systems".

Using only an SNMP Console

Monitoring consoles use an SNMP Master Agent to communicate with the Enterprise Manager SNMP Subagent. The SNMP Master Agent must be configured correctly before the Enterprise Manager SNMP Subagent can communicate over SNMP to the SNMP Master Agent.

The following is an example of the process for configuring the SNMP Master Agent on a UNIX platform.

> **Note:** SNMP with the Enterprise Manager SNMP Subagent is not supported by all platforms. Refer to the installation or configuration guide for your specific platform to identify differences in the configuration procedures and to confirm SNMP support.

Configuring the SNMP Master Agent

The necessary SNMP files are automatically installed when you install Oracle Management Agent. After installing Oracle Management Agent, edit the following files:

```
$ORACLE_HOME/sysman/snmp/peer/CONFIG.master
$ORACLE_HOME/sysman/snmp/peer/CONFIG.encap
$ORACLE_HOME/sysman/snmp/peer/start_peer
/etc/snmp/conf/snmpd.conf
```

CONFIG.Master (Peer Master Agent file)

To edit the CONFIG.master file, find the line beginning with MANAGER and change the IP address coded in this line to match the IPaddress of the machine where the SNMP Traps will be sent.

CONFIG.encap (Peer Encapsulator file)

To edit the CONFIG.encap file, find the line AGENT AT PORT. It normally reads AGENT AT PORT 1161 WITH COMMUNITY public. If you modify the port number from 1161, you must also modify the start_peer script.

start_peer Script

To edit the start_peer script, find the line NEW_SNMPD_PORT= and verify that it is using the same port number as listed in the CONFIG.encap file. Find the line NEW_ TRAPD_PORT= and verify the port number is different than NEW_SNMPD_PORT=.

snmpd.conf File

Add the following line to the file:

trap <hostname or ipaddress>

Replace the information in brackets with the actual hostname or IPaddress of the local host where the file is located.

Once you have configured the SNMP Master Agent to use only the SNMP console, you then need to configure the SNMP Console to receive external traps on the managed node.

For more comprehensive configuration information, see the documentation specific to your platform; SNMP configuration differs from platform to platform.

Starting the SNMP Agents

Make sure no SNMP components are running. The three main components are the master_peer, encap_peer, and snmpd. If any of these processes are running, use the ps command to find them, and the kill command to terminate these processes.

For example, the following checks to see if the SNMP Master Agent is running.:

```
ps -ef | grep snmp
```

Start the PEER Master Agent, PEER encapsulator, and native UNIX SNMP Agent:

```
  cd $ORACLE_HOME/network/snmp/peer
  su root
  ./start_peer -a
```

This command starts all three processes. Then use the ps command to determine if all three processes were started:

```
ps -ef |grep peer
ps -ef |grep snmpd
ps -ef |grep snmp
```

Configuring SNMP on IBM AIX Systems

Because IBM AIX has its own native SNMP agent, you do not use the start_peer script for SNMP Agent start up.

For IBM AIX systems:

1. Add the following line to the snmpd.conf file:

   ```
   smux 0.0 "" server_IP_address
   ```

   server_IP_address is the interface to which Oracle Management Agent is bound. To confirm this address, at the UNIX prompt, enter `hostname` and press RETURN. Then ping the returned host name and use the IP address that is returned.

2. To enable the SNMP Master Agent to read the updated snmpd.conf file, you must refresh or restart snmpd.

   To refresh:

```
refresh -s snmpd
```

To restart:

```
stopsrc -s snmpd
startsrc -s snmpd
```

3. Load the Oracle MIBs as described in "Loading the Oracle MIBs".

Loading the Oracle MIBs

Load the Oracle MIBs according to the instructions provided in your SNMP Console configuration guide.

Configuring SNMP with Enterprise Manager

The Enterprise Manager SNMP Subagent and necessary SNMP files are automatically installed when you install Oracle Management Agent.

> **Note:** Notification methods based on SNMP Traps must be configured by an administrator with super administrator privileges before any user can then choose to select one or more of these SNMP Trap methods while creating/editing a notification rule.

Controlling Enterprise Manager SNMP Subagent

The Enterprise Manager command line utility (emctl) is used to start, stop, or request the status of Enterprise Manager SNMP Subagent.

Use the following commands to start, stop, or request the status of the subagent. The following examples show the command and expected output.

Example 2–1 Starting the Enterprise Manager SNMP Subagent

```
% emctl start subagent

Oracle Enterprise Manager 11g Release 11.1.0.1.0
Copyright (c) 1996, 2010 Oracle Corporation. All rights reserved.
Starting sub agent ... started
```

Example 2–2 Stopping the Enterprise Manager SNMP Subagent

```
% emctl stop subagent

Oracle Enterprise Manager 11g Release 11.1.0.1.0
Copyright (c) 1996, 2010 Oracle Corporation. All rights reserved.
Stopping sub agent ... stopped
```

Example 2–3 Requesting Status of the Enterprise Manager SNMP Subagent

```
% emctl status subagent

Oracle Enterprise Manager 11g Release 11.1.0.1.0
Copyright (c) 1996, 2010 Oracle Corporation. All rights reserved.
Sub agent is running OR Sub agent is not running
```

Setting Up SNMP Trap Notifications

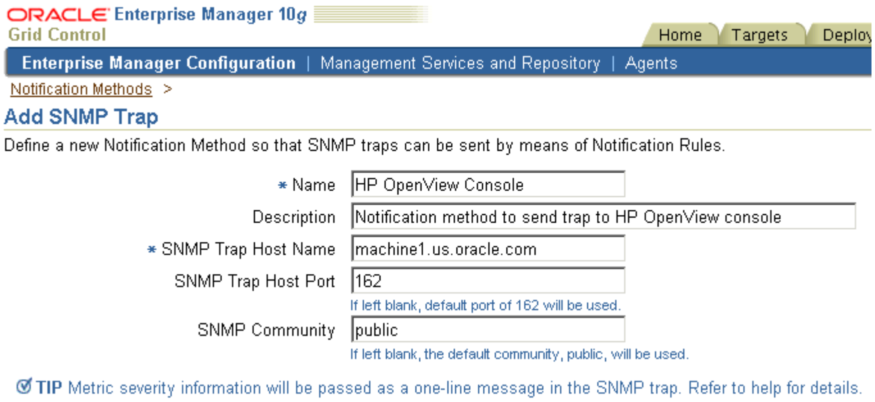
Follow these steps to set up SNMP Trap notifications:

- From the Enterprise Manager Grid Control Console, define an SNMP Trap destination by going to the Notification Methods page under **Setup**.

  Notification Methods allow you to define different mechanisms for sending notifications, for example, e-mail, SNMP Traps, running custom scripts, or all three. Once defined, these methods can then be used with Notification Rules for sending notifications to administrators as a result of alert occurrences.

- Under **Scripts and SNMP Traps**, select **SNMP Trap** from the **Add** menu.
- On the Add SNMP Trap page, provide the name of the host (machine) on which the SNMP Master Agent is running, the port it is listening on, and the community it is part of (Figure 2–2).

Figure 2–2 Adding SNMP Trap


ORACLE Enterprise Manager 10g
Grid Control
Home
Targets
Deploy
Enterprise Manager Configuration | Management Services and Repository | Agents
Notification Methods >

Add SNMP Trap

Define a new Notification Method so that SNMP traps can be sent by means of Notification Rules.

* Name	HP OpenView Console
Description	Notification method to send trap to HP OpenView console
* SNMP Trap Host Name	machine1.us.oracle.com
SNMP Trap Host Port	162 If left blank, default port of 162 will be used.
SNMP Community	public If left blank, the default community, public, will be used.

TIP Metric severity information will be passed as a one-line message in the SNMP trap. Refer to help for details.

> **Note:** A Test Trap button exists for you to test your setup.

- After you have defined the Notification Method, define a Notification Rule by going to the Notification Rules page under **Preferences**.

  Notification rules allow you to choose the targets and conditions for which you want to receive notifications from Enterprise Manager.

- Click **Create** on the Notification Rules page to display the Create Notification Rule property pages. See the online help provided for these pages and follow the instructions carefully to create your Notification Rule. While defining the Notification Rule, remember to include the Notification Method created in the previous steps.

The MIBs for each trap type are available in Appendix F, "Interpreting Variables of the Enterprise Manager MIB".

Example 2–4 shows a typical SNMP Trap.

Example 2–4 SNMP Trap

```
Tue Oct 28 05:00:02 2003

Command: 4
   Enterprise: 1.3.6.1.4.1.111.15.2
   Agent: 138.1.6.200
   Generic Trap: 6
   Specific Trap: 1
   Time Stamp: 8464:39.99
   Count: 11
Name: 1.3.6.1.4.1.111.15.1.1.1.2.1
   Kind: OctetString
   Value: "mydatabase"
Name: 1.3.6.1.4.1.111.15.1.1.1.3.1
   Kind: OctetString
   Value: "Database"
Name: 1.3.6.1.4.1.111.15.1.1.1.4.1
  Kind: OctetString
  Value: "myhost.com"
Name: 1.3.6.1.4.1.111.15.1.1.1.5.1
   Kind: OctetString
   Value: "Owner's Invalid Object Count"
Name: 1.3.6.1.4.1.111.15.1.1.1.6.1
   Kind: OctetString
   Value: "Invalid Object Owner"
Name: 1.3.6.1.4.1.111.15.1.1.1.7.1
   Kind: OctetString
   Value: "SYS"
Name: 1.3.6.1.4.1.111.15.1.1.1.8.1
   Kind: OctetString
   Value: "28-OCT-2003 04:59:10 (US/Eastern GMT)"
Name: 1.3.6.1.4.1.111.15.1.1.1.9.1
   Kind: OctetString
   Value: "Warning"
Name: 1.3.6.1.4.1.111.15.1.1.1.10.1
   Kind: OctetString
   Value: "12 object(s) are invalid in the SYS schema."
Name: 1.3.6.1.4.1.111.15.1.1.1.11.1
   Kind: OctetString
   Value: "Database Metrics"
Name: 1.3.6.1.4.1.111.15.1.1.1.12.1
   Kind: OctetString
   Value: "SYSMAN"
```

3

Oracle MIB Overview

This chapter covers the following topics:

- Oracle Products Providing SNMP Support
- MIBs Supporting These Products
- Interpreting SNMP OIDs

Oracle Products Providing SNMP Support

This manual documents four Oracle products that provide SNMP support. These products are:

- the Oracle Database
- the Oracle Listener
- Oracle Enterprise Manager

Oracle Database

The Oracle Server is the relational database management system (RDBMS) developed and sold by Oracle. This RDBMS includes the database that stores information, as well as the server software that accesses and manipulates that information.

For more information about Oracle Server, see the *Oracle Database Concepts* manual specific to your system.

Oracle Listener

The Listener is a protocol-independent application listener that receives connections on behalf of applications running over a variety of underlying protocols. The Listener is a component of Oracle Net Services, Oracle's remote data access product, enabling client/server and server/server communications across any network. It runs as a single process or task.

For more information about the Listener, see the manual *Oracle Database Net Services Administrator's Guide*.

Oracle Enterprise Manager

Oracle Enterprise Manager provides Oracle database administrators with an intuitive, easy-to-use interface and a high degree of automation through remote task execution and reactive and proactive management capabilities. It also scales to accommodate both very large database configurations, while being flexible enough to handle installations with many databases. Oracle Enterprise Manager includes an open-architecture design, providing third-party vendors and customers with the ability to integrate their applications into Oracle's system management platform.

MIBs Supporting These Products

The MIB supporting these Oracle products are:

- Network Services MIB
- Public RDBMS MIB
- Private Oracle Database MIB
- Other Oracle Private MIBs

Network Services MIB

The public Network Services MIB (as described in RFC 1565) contains generic variables designed to apply to all types of network service applications. Oracle has implemented those variables of this MIB that are relevant to its Oracle Database.

Specific values for these variables are retrieved from a variety of sources, including configuration files, tables internal to the network service application, and from the master agent itself.

The Listener does not make use of any Network Services MIB variables.

Public RDBMS MIB

The public RDBMS MIB is the proposed standard MIB for relational databases that has been defined by the IETF Working Group. This MIB allows for database discovery, identification of the database, and characterization of database size and activity level.

The public RDBMS MIB includes management variables that are both common to all RDBMSs and independent of vendor.

While this MIB makes a clear distinction between the database and its server, at this time the Oracle Database subagent only recognizes the standard Oracle configuration of one database to one server. Thus, it does not account for the Oracle Real Application Clusters server or gateways.

The information in this MIB is mostly retrieved from dynamic performance tables (V$ tables) and the INIT.ORA configuration file of the Oracle Database.

Private Oracle Database MIB

The private Oracle Database MIB contains additional RDBMS statistics that are specific to the Oracle database architecture. Like the public RDBMS MIB, the private Oracle Database MIB derives most of its information from dynamic performance tables of the database and from the INIT.ORA database configuration file.

The private Oracle Database MIB is platform-independent. This allows a DBA to use one MIB, regardless of the number of platforms being managed.

Other Oracle Private MIBs

In addition to the private database MIB, Oracle has defined private MIBs for the following products:

- Oracle Network Listener

- Oracle Enterprise Manager

Interpreting SNMP OIDs

This section briefly covers how SNMP object identifiers (OIDs) are assigned, as it pertains to the MIBs that Oracle has implemented. Specifically, this section covers the following topics:

- interpreting Oracle OIDs
- interpreting OIDs for Oracle-implemented public MIBs
- interpreting OIDs for instances of private Oracle MIB variables

For more information on SNMP OIDs, see any of the standard SNMP texts listed in the preface.

For easy reference, tables listing the object identifiers for each object of a given Oracle-implemented MIB can be found in the appendix covering that MIB.

Interpreting Oracle OIDs

The SNMP standard (RFC 1442) specifies that an object identifier (OID) be used to uniquely identify each object. An OID is a sequence of elements that indicates a hierarchical organization of identifiers. These elements take the form of a series of ″dotted″ integers, similar in format to an Internet address.

An example OID for an private Oracle MIB variable follows:

1.3.6.1.4.1.111.4.1.7.1.1

In this OID:

- The first element (1) refers to the *iso* object
- The second element (3) refers to the *org* object
- The third element (6) refers to the *dod* object
- The fourth element (1) refers to the *internet* object
- The fifth element (4) refers to the *private* object
- The sixth element (1) refers to the *enterprises* object
- The seventh element (111) refers to the Oracle object

Each of these elements is assigned by entities outside Oracle. Thus, all objects within the Oracle ID space (that is, those objects to which Oracle has assigned OIDs), share the root OID 1.3.6.1.4.1.111.

Continuing use of this OID example, Oracle has defined the elements of its private ID space as follows:

- The eighth element (4) is the MIB service identifier. Oracle has defined the following MIB service identifiers:
  - 4 - Oracle Private Database MIB
  - 5 - Oracle Listener MIB
- The ninth element (1) is the MIB object identifier. Oracle has defined the following MIB object identifiers for each service:
  - 1 - SNMP variable

  - 2 - SNMP trap
- The tenth element (7) is the MIB table identifier (oraDbConfigTable, in this case).
- The eleventh element (1) is the table entry identifier. This element can be considered as a place holder and is always 1.
- The twelfth element (1) is the leaf object identifier (identifying a particular variable in the table).

Interpreting OIDs for Oracle-Implemented Public MIBs

In addition to the four private Oracle MIBs that fall within Oracle OID space, Oracle is implementing portions of two public MIBs (Network Services MIB and RDBMS MIB) that fall outside Oracle OID space. The object IDs for the variables in these MIBs have been assigned in their respective RFCs.

An example OID for a public RDBMS MIB variable follows:

1.3.6.1.2.1.39.1.2.1.3

In this OID, the first four elements (1.3.6.1) match those for the Oracle root ID, indicating that this object falls under the iso, org, dod and internet objects. The fifth element (2), however, indicates that this object falls within the Internet management OID space. The seventh element (39) indicates that this object falls within the public RDBMS MIB. All variables of the public RDBMS MIB share this root OID of 1.3.6.1.2.1.39 The ninth element (2) indicates that this object is part of the rdbmsDbInfoTable. The eleventh element (3) identifies this as the leaf object rdbmsDbInfoSizeUnits.

An example OID for a public Network Services MIB variable follows:

1.3.6.1.2.1.27.1.1.1.6

In this OID, the first four elements (1.3.6.1) match those for the Oracle root ID, indicating that this object falls under the iso, org, dod and internet entities. The fifth element (2), however, indicates that this object falls within Internet management OID space. The seventh element (27), indicates that this object falls within the public Network Services MIB. All variables of the public Network Services MIB share this root OID of 1.3.6.1.2.1.27. The ninth element (1) indicates that this object is part of the applTable. The tenth element (6) identifies this as the leaf object applOperStatus.

Interpreting OIDs for Instances of Oracle MIB Variables

Because variables in Oracle’s MIBs are defined in tables, there can be multiple instances of a single variable. If, for example, there are two Oracle databases running on a given managed node, each database will have its own value for MIB variables such as applInboundAssociations, rdbmsDbName, and rdbmsSrvInfoDiskReads.

Not all variables are indexed on a per-service basis. It is also possible for a variable to have many instances for a single database. For example, while rdbmsSrvParamEntry describes a single database configuration parameter, that same managed node will have many instances of rdbmsSrvParamName.

To uniquely identify the multiple instances of these variables, each MIB table is indexed by one or more variables which, together, uniquely identify the rows of the table. (These index variables are conceptually equivalent to the primary key of a relational database table.) To refer to a particular instance of a variable, concatenate the variable’s OID with the values of the index variables of the MIB table to which the variable belongs.

For instance, rdbmsDbName is defined within the rdbmsDbTable, which is indexed by the variable rdbmsDbIndex. For example, assume two databases are running on a host, one with SNMP index 2, the other with SNMP index 4. Then the name of the first database can be specified by concatenating the OID for rdbmsDbName (1.3.6.1.2.1.39.1.1.1.4) with the appropriate value of rdbmsDbIndex (2), or 1.3.6.1.2.1.39.1.1.1.4.2. Similarly, the name of the second database is the value of 1.3.6.1.2.1.39.1.1.1.4.4.

If a table is indexed by more than one variable, add the appropriate value of each index variable to the end of the OID, in the order they are listed in the table's MIB definition INDEX clause, separated by dots. The size (oraDbDataFileSizeAllocated, or 1.3.6.1.4.1.111.4.1.3.1.3) of the fifth data file (5) of the second database on this host (whose rdbmsDbIndex is 4) is the value of 1.3.6.1.4.1.111.4.1.3.1.3.4.5.

4

Reading the MIB Variable Descriptions

This chapter helps you understand how to read MIB variable descriptions. It covers the following:

- Variable Name Description
- MIB Definition

Note that the STATUS element of SNMP MIB definition, Version 2, is not included in these MIB variable descriptions. Since Oracle has implemented all MIB variables as CURRENT, this value does not vary.

Variable Name Description

Syntax

Maps to the SYNTAX element of SNMP MIB definition, Version 2.

Max-Access

Maps to the MAX-ACCESS element of SNMP MIB definition, Version 2.

Status

Maps to the STATUS element of SNMP MIB definition, Version 2.

Explanation

Describes the function, use and precise derivation of the variable. (For example, a variable might be derived from a particular configuration file parameter or performance table field.) When appropriate, incorporates the DESCRIPTION part of the MIB definition, Version 2.

Typical Range

Describes the typical, rather than theoretical, range of the variable. For example, while integer values for many MIB variables can theoretically range up to 4294967295, a typical range in an actual installation will vary to a lesser extent. On the other hand, some variable values for a large database can actually exceed this “theoretical” limit (a “wraparound”). Specifying that a variable value typically ranges from 0 to 1,000 or 1,000 to 3 billion will help the third-party developer to develop the most useful graphical display for the variable.

Significance

Describes the significance of the variable when monitoring a typical installation. Alternative ratings are Very Important, Important, Less Important, or Not Normally Used. Clearly, the DBA will want to monitor some variables more closely than others. However, which variables fall into this category can vary from installation to installation, depending on the application, the size of the database, and on the DBA's objectives. Nevertheless, assessing a variable's significance relative to the other variables in the MIB can help third-party developers focus their efforts on those variables of most interest to the most DBAs.

Related Variables

Lists other variables in this MIB, or other MIBs implemented by Oracle, that relate in some way to this variable. For example, the value of this variable might derive from that of another MIB variable. Or perhaps the value of this variable varies inversely to that of another variable. Knowing this information, third-party developers can develop useful graphic displays of related MIB variables.

Suggested Presentation

Suggests how this variable can be presented most usefully to the DBA using the management application: as a simple value, as a gauge, or as an alarm, for example.

MIB Definition

Example 4–1 shows a typical MIB definition used by Enterprise Manager.

Example 4–1 MIB Definition

```
ORACLE-ENTERPRISE-MANAGER-4-MIB DEFINITIONS ::= BEGIN
IMPORTS
TRAP-TYPE
FROM RFC-1215
DisplayString
FROM RFC1213-MIB
OBJECT-TYPE
FROM RFC-1212
enterprises
FROM RFC1155-SMI;

oracle OBJECT IDENTIFIER ::= { enterprises  111 }
oraEM4 OBJECT IDENTIFIER ::= { oracle  15 }
oraEM4Objects OBJECT IDENTIFIER ::= { oraEM4  1 }
oraEM4AlertTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF OraEM4AlertEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
   "Information on alerts generated by Oracle Enterprise Manager. This table is
not queryable; it exists only to document the variables included in the
oraEM4Alert trap.  Each trap contains a single instance of each variable in the
table."
    ::= { oraEM4Objects  1 }

oraEM4AlertEntry OBJECT-TYPE
    SYNTAX  OraEM4AlertEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
     "Information about a particular Oracle Enterprise Manager alert."
    INDEX   { oraEM4AlertIndex }
    ::= { oraEM4AlertTable  1 }

OraEM4AlertEntry ::=
    SEQUENCE {
        oraEM4AlertIndex
            INTEGER,
        oraEM4AlertTargetName
   DisplayString,

        oraEM4AlertTargetType
   DisplayString,

        oraEM4AlertHostName
   DisplayString,

        oraEM4AlertMetricName
   DisplayString,

        oraEM4AlertKeyName
   DisplayString,

        oraEM4AlertKeyValue
   DisplayString,

        oraEM4AlertTimeStamp
   DisplayString,

        oraEM4AlertSeverity
   DisplayString,

        oraEM4AlertMessage
   DisplayString,

        oraEM4AlertRuleName
   DisplayString

        oraEM4AlertRuleOwner
   DisplayString
     }
oraEM4AlertIndex OBJECT-TYPE
    SYNTAX  INTEGER
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "Index of a particular alert, unique only at the moment an alert is
generated."
    ::= { oraEM4AlertEntry  1 }

oraEM4AlertTargetName OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The name of the target to which this alert applies."
    ::= { oraEM4AlertEntry  2 }

oraEM4AlertTargetType OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The type of the target to which this alert applies."
    ::= { oraEM4AlertEntry  3 }
```

```
oraEM4AlertHostName OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The name of the host on which this alert originated."
    ::= { oraEM4AlertEntry  4 }

oraEM4AlertMetricName OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The name of the metric which generated this alert."
    ::= { oraEM4AlertEntry  5 }

oraEM4AlertKeyName OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The name of the key-column, if present, for the metric which generated this
alert."
    ::= { oraEM4AlertEntry  6 }

oraEM4AlertKeyValue OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The value of the key-column, if present, for the metric which generated this
alert."
    ::= { oraEM4AlertEntry  7 }

oraEM4AlertTimeStamp OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The time at which this alert was generated."
    ::= { oraEM4AlertEntry  8 }

oraEM4AlertSeverity OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The severity of the alert e.g. Critical."
    ::= { oraEM4AlertEntry  9 }

oraEM4AlertMessage OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The message associated with the alert."
    ::= { oraEM4AlertEntry  10 }

oraEM4AlertRuleName OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The name of the notification rule that caused this notification."
    ::= { oraEM4AlertEntry  11 }

oraEM4AlertRuleOwner OBJECT-TYPE
    SYNTAX  DisplayString
    ACCESS  read-only
    STATUS  mandatory
    DESCRIPTION
     "The owner of the notification rule that caused this notification."
    ::= { oraEM4AlertEntry  12 }

oraEM4Traps OBJECT IDENTIFIER ::= { oraEM4  2 }

oraEM4Alert TRAP-TYPE
    ENTERPRISE  oraEM4Traps
    VARIABLES   { oraEM4AlertTargetName, oraEM4AlertTargetType,
                  oraEM4AlertHostName, oraEM4AlertMetricName,
                  oraEM4AlertKeyName, oraEM4AlertKeyValue, oraEM4AlertTimeStamp,
                  oraEM4AlertSeverity, oraEM4AlertMessage,
                  oraEM4AlertRuleName, oraEM4AlertRuleOwner }
    DESCRIPTION
     "The variables included in the oraEM4Alert trap."
    ::= 1
END
```

5

Designing Management Applications Based on Oracle MIBs

This chapter covers the following topics:

Using MIB Variables for Different Purposes

MIB variables can be used for two different purposes. If they are used for fault management, the DBA is interested in significant changes in a variable's value or in a ratio over a short period of time. Once the DBA becomes aware of that change, its cause can be determined and corrected, if necessary. However, MIB variables can also be used to tune performance. In this case, the goal is to establish benchmark performance, as measured by the appropriate variable(s) or ratio(s).

Using a Sufficient Sample Size to Calculate Tuning Ratios

When MIB variables are used for performance tuning, make sure that you poll enough instances of a variable to measure performance with some certainty. Polling 1,000 separate statistics is considered acceptable for any database instance performance ratio mentioned in this chapter. When designing graphic presentations for these ratios, consider alerting the user when less than 1,000 separate instances of a variable have been used to calculate a tuning ratio.

Presenting Performance Ratios Graphically

In general, users will be most interested in monitoring changes in performance ratios over time. Thus, a basic time series graph that plots changes in the values of these ratios against time is very useful.

Scaling MIB Variable Values Dynamically

The scalar values for MIB variables (and ratios based on these variables) measuring database performance can vary tremendously, depending on the type of database application, the number of users accessing the database, and the amount of data being processed. This fact poses a challenge to the developer of a management application presenting this information. Graphic elements intended to accommodate the full range of scalar values can mislead when presenting scalar values at the low end of the range; even large changes relative to their size can appear insignificant when plotted on a graph designed to accommodate much greater values achievable at other installations.

One good way to accommodate users at both ends of the data spectrum is to scale the presentation of data dynamically. This involves adjusting the scale of the graph measuring the data as the data attains certain thresholds. For example, if you were to plot against time the speed of a jet from takeoff to its cruising level, you might set the upper limit of the graph upon takeoff at 200 mph; when that threshold is reached, you could dynamically adjust the upper limit to 400 mph, and then adjust it twice more as thresholds of 400 and 600 mph are attained.

Working Around the Wraparound

Given the tremendous processing power available today, and given the enormous upper limit to the size of an Oracle database, it may not be surprising that the values of some MIB variables do in fact exceed (wrap around) the SNMP-defined upper limit for 32-bit counter and integer (4294967295) values. A management application designer can best work around this problem by always showing instance startup time (applUpTime) to the user, and encouraging the user to poll the values of such variables frequently enough to determine whether a wrap around has occurred. A minimum of once every one-half hour for “wrap around candidates“ is suggested.

If the value of a counter variable has decreased

, even though the database instance has continued to run, one can assume that the value has wrapped around the upper limit of the range. However, one must poll frequently enough to ensure that the delta of the value is less than the theoretical limit of the range. The values and ratios you present to the user can then take the wraparound into account.

Most Useful Database Instance Performance Ratios

This section describes those ratios that are most useful in tuning the performance of an Oracle database instance. These are the performance measures of greatest interest to potential customers of management applications for the Oracle database. Each of these ratios is based on variables of the private Oracle Database MIB. These ratios are listed alphabetically, not in order of importance.

For more information on these ratios, see the *Oracle Database Performance Tuning Guide* specific to your system. For information on the MIB variables that are used to calculate these ratios, see Appendix B, "Interpreting Variables of the Oracle Database MIB".

Block Changes Per Transaction

This ratio measures the amount of Data Manipulation Language (DML) work that each transaction performs. Creating or dropping indexes impacts this value, because changes to index blocks increment it.

```
oraDbSysDbBlockChanges / oraDbSysUserCalls
```

Block Get Rate

This ratio determines the block get rate. The block get rate is a basic measure of the rate at which the application system references the database. The time unit typically used in this ratio is one second.

```
(oraDbSysConsistentGets + oraDbSysDbBlockGets)/ time unit
```

Block Visits Per Transaction

This ratio measures the work database load imposed for each transaction; if it is moving independently, then this strongly indicates that there has been a change in the application workload.

```
(oraDbSysDbBlockGets + oraDbSysConsistentGets) / oraDbSysUserCommits
```

Cache Hit Ratio

This ratio measures the effectiveness of the buffer cache. The normally acceptable range is 70 - 85%.

```
(oraDbSysConsistentGets + oraDbSysDbBlockGets - oraDbSysPhysReads) / (oraDbSysConsistentGets
+ oraDbSysDbBlockGets)
```

Call Rate

This ratio measures the work demand rate being placed on the instance from all work sources. It should be noted, however, that this rate may not be directly comparable across application system version changes where row at a time loop constructs have been recoded as set operations or vice versa. Use of the array interface will also affect this ratio.

```
(oraDbSysRecursiveCalls + oraDbSysUserCalls)/ time unit
```

Calls Per Transaction

This ratio measures the number of client requests made for each transaction. Calls for each transaction can be used to detect changes in the application, or in the ways in which it is being used. This value may rise sharply as ad hoc queries increase.

```
oraDbSysUserCalls / oraDbSysUserCommits
```

Changed Block Ratio

This ratio measures the balance between queries and DML within this database application. Changes in this ratio indicate and/or quantify changes in indexation or application usage.

```
oraDbSysDbBlockChanges / (oraDbSysDbBlockGets + oraDbSysConsistentGets)
```

Consistent Change Ratio

This ratio measures the extent to which applications are having to exercise the read consistency mechanism. In this connection, it is important to realize that the query processing parts of UPDATE and/or DELETE operations are subject to read consistency.

```
oraDbSysConsistentChanges / oraDbSysConsistentGets
```

Continued Row Ratio

This ratio should be very close to zero, except in applications handling long LONG columns. If this ratio increases over time, usually PCTFREE has been set too low on one or more tables.

```
oraDbSysTableFetchContinuedRow / (oraDbSysTable FetchRowid + oraDbSysTableScanRows)
```

Library Cache Miss Ratio

If this ratio begins to rise, then resource usage can be expected to increase. A rising library cache miss ratio may be due to wider use of application functionality causing more SQL statements and stored procedures to be active than had previously been the case.

```
oraDbLibraryCacheReloads) / oraDbLibraryCachePins
```

Recursive to User Call Ratio

Under Oracle version 7 and Oracle version 8, a change in this ratio can reflect an application change, or indicate a need to adjust the size of the shared pool. Any marked change in the DDL load also affects this ratio.

```
oraDbSysRecursiveCalls / oraDbSysUserCalls
```

Redo Log Space Wait Ratio

This ratio measures memory allocation. If it is greater than 1 / 5,000, then the redo log buffer should be increased until the redo log space wait ratio stops failing.

```
oraDbSysRedoLogSpaceRequests / oraDbSysRedoEntries
```

Row Source Ratio

This ratio measures the percentage of the total rows retrieved which came from full table scans. As soon as this percentage starts to rise much above 0, the interpretation of other statistics may need to be reviewed.

```
oraDbSysTableScanRows / (oraDbSysTableFetchRowid + oraDbSysTableScanRows)
```

Sort Overflow Ratio

oraDbSysSortsDisks / (oraDbSysSortsMemory + oraDbSysSortsDisks) is the sort overflow ratio. This ratio yields the ratio of the number of sorts which are using temporary segments. Under restricted circumstances when there is a predominance of medium size sorts, increasing the sort area size may be effective.

```
oraDbSysSortsDisks / (oraDbSysSortsMemory + oraDbSysSortsDisks)
```

Transaction Rate

The transaction rate is a basic measure of application work, and would be calibrated in transactions per second (tps) for a typical OLTP benchmark. Administrators should be particularly concerned if a fall in this value is associated with a rise in the number of connected users or vice versa. Changes in application structure or work patterns can also distort this figure.

```
oraDbSysUserCommits
```

User Call Rate

This rate measures the work demand rate being posed by client side applications running under the instance. It should be noted, however, that this may not be directly comparable across application system version changes where code has been moved from client to server side or vice versa.

```
oraDbSysUserCalls
```

User Calls Per Parse

This ratio indicates how well the application is managing its context areas. If it changes, then application change is the most likely explanation, but it may also indicate that usage patterns are changing and users are moving from one module to another either more frequently or less frequently.

Although the shared SQL area makes the maximizing of this ratio less important than with earlier versions of Oracle, it is still possible to reduce resource usage by raising this ratio.

```
oraDbSysUserCalls / oraDbSysParseCount
```

User Rollback Ratio

oraDbSysUserRollbacks / (oraDbSysUserCommits + oraDbSysUserRollbacks) is the user rollback ratio. The user rollback ratio indicates the rate at which application transactions are failing. Rolling back a transaction uses significant resources, and would seem to indicate that all of the resources expended in executing the transaction have been wasted.

```
oraDbSysUserRollbacks / (oraDbSysUserCommits + oraDbSysUserRollbacks)
```

Part II

List of Appendices

Part II covers appendixes that serve as reference materials for developers and administrators.

Part II contains the following appendixes:

A

Configuration Files

This appendix discusses the configuration files that are generated by Oracle Management Agent, and the parameters that can be set to optimize agent operation for different system setups. The following topics are discussed:

- Configuration Files
- User-configurable Parameters

Configuration Files

The following files control the operation of the Oracle Management Agent.

snmp_ro.ora

The snmp_ro.ora file is located in $ORACLE_HOME/network/admin or a directory specified by the TNS_ADMIN environment variable. Do NOT update this read-only file.

snmp_rw.ora

The snmp_rw.ora is located in the Agent's $ORACLE_HOME\network\admin. You can modify this read/write file, but this should be done carefully.

User-configurable Parameters

These parameters are used in the snmp_rw.ora configuration file.

dbsnmp.polltime = nn

The time interval (seconds) that the agent polls the database to check whether it is down. If the database has gone down or was never connected, this is the interval between retries. The default is 30 seconds.

> **Note:** If Oracle Management Agent must monitor more than two instances, you should increase the value of DBSNMP.POLLTIME proportionally with the number of monitored instances.
>
> For example:
>
> The Oracle Management Agent needs to monitor 10 instances. DBSNMP.POLLTIME should be set to 150. (10/2 * 30 = 150)

dbsnmp.noheuristic = {true/false}

The value of this parameter determines whether the Oracle Management Agent will use a connection heuristic to ascertain the state of a monitored database (whether the database is up or down). By default, this value is set to FALSE (Agent uses the heuristic).

> **Note:** If the monitored target is a Real Application Clusters database instance, DBSNMP.NOHEURISTIC must be set to TRUE since the heuristic does not work against Real Application Clusters database instances.

B

Interpreting Variables of the Oracle Database MIB

This appendix provides information to help you interpret individual variables in the private Oracle Database MIB. Specifically, it includes sections covering SNMP variables in the following tables:

- oraDbSysTable
- oraDbTablespaceTable
- oraDbDataFileTable
- oraDbLibraryCacheTable
- oraDbLibraryCacheSumTable
- oraDbSGATable
- oraDbConfigTable

For information about the database instance performance ratios listed in this appendix, see the section Most Useful Database Instance Performance Ratios in Chapter 5, "Designing Management Applications Based on Oracle MIBs". For more information about the Oracle database server and specific topics covered in this appendix, see the *Oracle Database Reference* and the *Oracle Database Concepts* specific to your system.

oraDbSysTable

The oraDbSysTable contains various measurements used to support database activities of current sessions on the managed node. This information has been found to be particularly useful for monitoring global database instance performance.

Each entry in the oraDbSysTable represents a current database instance on the node. Each entry is indexed by rdbmsDb Index. However, because rdbmsDbTable contains an entry for each database configured for SNMP support, whether or not it is actively open, there may or may not be a one-to-one correspondence between entries in the two tables at any point in time. (For details about this index, see Appendix D, “Interpreting Variables of the Public RDBMS MIB.”) The value of each oraDbSysTable variable is retrieved directly from a corresponding field in the V$SYSSTAT table for that database instance on the managed node.

In many cases, a variable is meaningful only when considered in conjunction with other variables, often as a ratio. For more information on the ratios mentioned in the following variable descriptions, and for their significance in tuning the Oracle database server, see the *Oracle Database Performance Tuning Guide* specific to your system.

Table B–1 lists each variable of the oraDbSysTable and its corresponding object ID.

Table B–1 oraDbSysTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbSysConsistentChanges	1.3.6.1.4.1.111.4.1.1.1.1
oraDbSysConsistentGets	1.3.6.1.4.1.111.4.1.1.1.2
oraDbSysDbBlockChanges	1.3.6.1.4.1.111.4.1.1.1.3
oraDbSysDbBlockGets	1.3.6.1.4.1.111.4.1.1.1.4
oraDbSysFreeBufferInspected	1.3.6.1.4.1.111.4.1.1.1.5
oraDbSysFreeBufferRequested	1.3.6.1.4.1.111.4.1.1.1.6
oraDbSysParseCount	1.3.6.1.4.1.111.4.1.1.1.7
oraDbSysPhysReads	1.3.6.1.4.1.111.4.1.1.1.8
oraDbSysPhysWrites	1.3.6.1.4.1.111.4.1.1.1.9
oraDbSysRedoEntries	1.3.6.1.4.1.111.4.1.1.1.10
oraDbSysRedoLogSpaceRequests	1.3.6.1.4.1.111.4.1.1.1.11
oraDbSysRedoSyncWrites	1.3.6.1.4.1.111.4.1.1.1.12
oraDbSysSortsDisk	1.3.6.1.4.1.111.4.1.1.1.13
oraDbSysSortsMemory	1.3.6.1.4.1.111.4.1.1.1.14
oraDbSysSortsRows	1.3.6.1.4.1.111.4.1.1.1.15
oraDbSysTableFetchRowid	1.3.6.1.4.1.111.4.1.1.1.16
oraDbSysTableFetchContinuedRow	1.3.6.1.4.1.111.4.1.1.1.17
oraDbSysTableScanBlocks	1.3.6.1.4.1.111.4.1.1.1.18
oraDbSysTableScanRows	1.3.6.1.4.1.111.4.1.1.1.19
oraDbSysTableScansLong	1.3.6.1.4.1.111.4.1.1.1.20
oraDbSysTableScansShort	1.3.6.1.4.1.111.4.1.1.1.21
oraDbSysUserCalls	1.3.6.1.4.1.111.4.1.1.1.22
oraDbSysUserCommits	1.3.6.1.4.1.111.4.1.1.1.23
oraDbSysUserRollbacks	1.3.6.1.4.1.111.4.1.1.1.24

A description of each of these variables follows.

oraDbSysConsistentChanges

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for CONSISTENT CHANGES in V$SYSSTAT. It is incremented each time that a consistent get operation cannot accept the block in its current state, owing to a change made since the query consistency point.

oraDbSysConsistentChanges is used to calculate the following database instance performance ratio: the consistent change ratio.

Typical Range

two or more orders of magnitude less than the value of oraDbSysConsistentGets

Significance

important

Related Variables

oraDbSysConsistentGets

Suggested Presentation

counter

oraDbSysConsistentGets

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for CONSISTENT GETS in V$SYSSTAT. This value is incremented each time that the Oracle Buffer Manager is requested to locate a database block as part of a read consistent operation (a query). Only if the requested block is not found in Oracle's Buffer Cache will consistent get result in physical I/O.

oraDbSysConsistentGets is used to calculate the following database instance performance ratios: block get rate, block visits per transaction, cache hit ratio, changed block ratio, and the consistent change ratio.

Typical Range

0 - 4294967295; can increment at several thousand per second in the largest systems, so this variable is definitely a "wraparound" candidate

Significance

very important

Related Variables

oraDbSysUserCommits, oraDbSysConsistentChanges, oraDbSysPhysReads, oraDbSysDbBlockChanges, oraDbSysDbBlockGets

Suggested Presentation

counter

oraDbSysDbBlockChanges

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for DB BLOCK CHANGES in V$SYSSTAT. It is incremented each time that a database block is in memory to create a "dirty block." This statistic is a rough indication of total database work.

oraDbSysDbBlockChanges is used to calculate the following database instance performance ratios: block changes per transaction, and the changed block ratio.

Typical Range

0 - 4294967295; can increment at several thousand per second in the largest systems, so this variable is definitely a "wraparound" candidate; value tends to be lower than that for oraDbSysConsistentGets

Significance

one of the lesser measures of work rate or throughput

Related Variables

oraDbSysDbBlockGets, oraDbSysConsistentGets, oraDbSysUserCalls

Suggested Presentation

counter

oraDbSysDbBlockGets

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for DB BLOCK GETS in V$SYSSTAT. It is incremented each time the buffer manager is asked for a database block in current mode, that is, regardless of read consistency. This mode is used to locate blocks which are to be updated, and also to access control information such as segment headers. Only if the requested block is not found in Oracle's Buffer cache will a db block get result in physical I/O.

oraDbSysDbBlockGets is used to calculate the following database instance performance ratios: block get rate, block visits per transaction, the cache hit ratio, and the changed block ratio.

Typical Range

0 - 4294967295; can increment at several thousand per second in the largest systems, so this variable is definitely a “wraparound” candidate.

Significance

very important

Related Variables

oraDbSysDbBlockChanges, oraDbSysConsistentGets, oraDbSysPhysReads, oraDbSysUserCalls

Suggested Presentation

counter

oraDbSysFreeBufferInspected

Syntax

Counter32

Max-Access

read-only

Explanation

The “free buffer inspected” parameter from V$SYSTAT.

Typical Range

0 - 4294967295

Significance

This indicates the number of reusable buffers skipped over from the end of the Least Recently Used (LRU) queue in order to find a buffer. The difference between this and “dirty buffers inspected” equals the number of buffers that could not be used because they were busy or needed to be written after rapid aging out. They may have a user, a waiter, or be being read or written.

Related Variables

oraDbSysFreeBufferRequested

Suggested Presentation

counter

oraDbSysFreeBufferRequested

Syntax
Counter32

Max-Access
read-only

Explanation
The “free buffer requested” parameter from V$SYSTAT.

Typical Range
0 - 4294967295

Significance
This is the count of the number of times a free buffer was requested to create or load a block.

Related Variables
oraDbSysFreeBufferInspected

Suggested Presentation
counter

oraDbSysParseCount

Syntax
Counter32

Max-Access
read-only

Explanation
This variable is retrieved directly from the value for PARSE COUNT(total) in V$SYSSTAT. It counts the number of parse calls received by the RDBMS from both connected applications and recursive calls. In Oracle, parse calls are frequently satisfied by reference to the Shared SQL area and this statistic does not measure the number of parses which have taken place, only the number of calls.

oraDbSysParseCount is used to calculate the following database instance performance ratio: user calls per parse.

Typical Range
0 - 4294967295; unlikely to increment at a rate greater than a few hundred per second

Significance
important

Related Variables

oraDbLibraryCacheTable variables, oraDbSysUserCalls

Suggested Presentation

counter

oraDbSysPhysReads

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for PHYSICAL READS in V$SYSSTAT. It increments each time a DB Block is retrieved from the disk subsystem, but is not necessarily equal to the number of I/O requests made to the host operating system, as multiblock I/O may be used. A physical read can be viewed as an SGA Buffer Cache miss, hampering throughput (contention induction) and CPU (code pathlength) processing.

oraDbSysPhysReads is used to calculate the following database instance performance ratio: the cache hit ratio.

Typical Range

0 - 4294967295; unlikely to increment at a rate greater than a few hundred per second

Significance

very important

Related Variables

oraDbSysConsistentGets, oraDbSysDbBlockGets

Suggested Presentation

counter

oraDbSysPhysWrites

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for PHYSICAL WRITES in V$SYSSTAT. It increments each time a DB Block is written to disk. All writes are performed either by DBWR or LGWR, two background processes responsible for getting Blocks out of the SGA and back on disk.

Typical Range

0 - 4294967295; can increment at several thousand per second in the largest systems, so this variable is definitely a “wraparound” candidate

Significance

important

Related Variables

oraDbSysRedo family

Suggested Presentation

counter

oraDbSysRedoEntries

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for REDO ENTRIES in V$SYSSTAT. Each time that REDO data is copied into the log buffer, this counter is incremented. The value varies in relation to how many changes are being made.

oraDbSysRedoEntries is used to calculate the following database instance performance ratio: redo log space wait ratio.

Typical Range

0 - 4294967295; can increment at several thousand per second in the largest systems, so this variable is definitely a “wraparound” candidate.

Significance

important

Related Variables

oraDbSysRedoLogSpaceRequests, oraDbSysRedoSyncWrites, oraDbSysUserCommits

Suggested Presentation

counter

oraDbSysRedoLogSpaceRequests

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for REDO LOG SPACE REQUESTS in V$SYSSTAT. This counts the number of times that a server process had to wait to acquire an entry in the redo log.

oraDbSysRedoLogSpaceRequests is used to calculate the following database instance performance ratios: redo log space wait ratio.

Typical Range

0 - 10,000,000 (a value of 0 is ideal, but an increment of 1 or 2 per second might be acceptable)

Significance

important

Related Variables

oraDbSysRedoEntries

Suggested Presentation

counter

oraDbSysRedoSyncWrites

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for REDO SYNC WRITES parameter in V$SYSSTAT. It increments when a process waits for a ReDo write to complete, typically as a result of a Commit.

Typical Range

0 - 10,000,000 (typically increments at less than 1 per second)

Significance

less important

Related Variables

oraDbSysRedoEntries, oraDbSysRedoLogSpaceRequest, oraDbSysUserCommits

Suggested Presentation

counter

oraDbSysSortsDisk

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for SORTS DISKS in V$SYSSTAT. It increments for each sort which has to allocate disk work space.

oraDbSysSortsDisks is used to calculate the following database instance performance ratio: the sort overflow ratio.

Typical Range

for OLTP it should be 0

Significance

important

Related Variables

oraDbSysRedoEntries, oraDbSysRedoLogSpaceRequest, oraDbSyncWrites, oraDbSysSortsMemory, oraDbUserCommits

Suggested Presentation

counter

oraDbSysSortsMemory

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for SORTS MEMORY in V$SYSSTAT. It is incremented for each sort which does not require the allocation of work space on disk.

oraDbSysSortsMemory is used to calculate the following database instance performance ratio: the sort overflow ratio.

Typical Range

0 - 4294967295; unlikely to increment faster than tens per second, except in the largest systems

Significance

important

Related Variables

oraDbSysRedoEntries, oraDbSysRedoLogSpaceRequest, oraDbSyncWrites, oraDbSysSortsDisks, oraDbUserCommits

Suggested Presentation

counter

oraDbSysSortsRows

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for SORTS ROWS in V$SYSSTAT. It is incremented for each sort, regardless of whether the allocation of disk space was required.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbSysRedoEntries, oraDbSysRedoLogSpaceRequest, oraDbSyncWrites, oraDbSysSortsMemory, oraDbSortsDisks, oraDbUserCommits

Suggested Presentation

counter

oraDbSysTableFetchRowid

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for TABLE FETCH BY ROWID in V$SYSSTAT. This counter increments when a row is fetched using a RowID (usually recovered from an index). It should be noted that this statistic records all rows visited by the database engine, whether or not they were return to the application.

oraDbSysTableFetchRowid is used to calculate the following database instance performance ratios: the continued row ratio and the row source ratio.

Typical Range

0 - 4294967295; can increment at several tens of thousands per second in the largest systems, so this variable is definitely a “wraparound” candidate

Significance

important

Related Variables

oraDbSysRedoEntries, oraDbSysRedoLogSpaceRequest, oraDbSyncWrites, oraDbSysSortsDisks, oraDbSysTableFetchContinuedRow, oraDbSysTableScanRows, oraDbUserCommits

Suggested Presentation

counter

oraDbSysTableFetchContinuedRow

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for TABLE FETCH CONTINUED ROW in V$SYSSTAT. It may be incremented either because a row has been encountered which spans more than a DB block, or because a migrated row has been retrieved by rowid.

oraDbSysTableFetchContinuedRow is used to calculate the following database instance performance ratio: the continued row ratio.

Typical Range
close to 0 unless long LONG values are in use.

Significance
important

Related Variables
oraDbSysTable FetchRowid, oraDbSysTableScanRows

Suggested Presentation
counter

oraDbSysTableScanBlocks

Syntax
Counter32

Max-Access
read-only

Explanation
This variable is retrieved directly from the value for TABLE SCAN BLOCKS GOTTEN in V$SYSSTAT. During scanning operations, the RDBMS retrieves each row sequentially. This counter increments each time the RDBMS steps into a new DB Block during the scan. oraDbSysTableScanBlocks indicates how many DB Blocks were consistently read from the Buffer Cache for scanning. Comparing this to the value of oraDbSysConsistentGets suggests how much Consistent Read activity can be attributed to scanning.

Typical Range
0 - 3 billion; in OLTP very low; in DSS applications, a few thousand per second, so this variable is definitely a “wraparound” candidate.

Significance
important

Related Variables
oraDbSysConsistentGets

Suggested Presentation
counter

oraDbSysTableScanRows

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for TABLE SCAN ROWS GOTTEN in V$SYSSTAT. This counter increases each time a row is processed during a Scan operation. Note that the row need not be parsed back to the calling application in order for it to be counted here.

oraDbSysTableScanRows is used to calculate the following database instance performance ratio: the row source ratio, and the continued row ratio.

Typical Range

0 - 100,000,000 per day

Significance

important

Related Variables

oraDbSysTableFetchContinuedRow, oraDbSysTableFetchRowid

Suggested Presentation

counter

oraDbSysTableScansLong

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for TABLE SCANS LONG TABLES in V$SYSSTAT. Long tables are those whose number of data blocks below the high water mark exceeds the value of the initialization parameter SMALL _TABLE_THRESHOLD. DB Blocks visited in Long Full Table Scans are not prompted in the LRS List.

Typical Range

in OLTP, typically 0

Significance

important

Related Variables

oraDbSysTableScansShort

Suggested Presentation

counter

oraDbSysTableScansShort

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for TABLE SCANS SHORT TABLES in V$SYSSTAT. Short tables can be defined as tables that do not meet the long table criteria, as defined by oraDbSysTableScansLong. Short tables are those whose number of data blocks above the low water mark is low than the value of the initialization parameter LONG_TABLE_THRESHOLD.

Typical Range

0 - 4294967295; can increment at a few hundred per second

Significance

important

Related Variables

oraDbSysTableScansLong

Suggested Presentation

counter

oraDbSysUserCalls

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for USER CALLS in V$SYSSTAT. A user call is a Parse, an Execute, or a Fetch. With the Oracle server's deferred and bundled execution capabilities, this statistic will not necessarily correlate well with client/server message traffic.

oraDbSysUserCalls is used to calculate the following database instance performance ratios: block changes per transaction, the call rate, calls per transaction, the recursive-to-user call ratio, the user call rate, and user calls per parse.

Typical Range

0 - 10,000,000 per day

Significance

very important

Related Variables

oraDbSysDbBlockChanges, oraDbSysParseCount, oraDbSysUserCommits

Suggested Presentation

counter

oraDbSysUserCommits

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for USER COMMITS in V$SYSSTAT. It records the number of database transactions successfully completed and records the number of aborted database transactions.

oraDbSysUserCommits is used to calculate the following database instance performance ratios: block visits per transaction, calls per transaction, and the transaction rate.

Typical Range

In most real world applications, the value is less the 1 per second.

Significance

In OLTP, very important

Related Variables

oraDbSysDbBlockGets, oraDbSysConsistentGets, oraDbUserCalls

Suggested Presentation

counter

oraDbSysUserRollbacks

Syntax

Counter32

Max-Access

read-only

Explanation

This variable is retrieved directly from the value for USER ROLLBACKS in V$SYSSTAT. It records the number of aborted database transactions.

oraDbSysUserRollbacks is used to calculate the following database instance performance ratio: the user rollback ratio.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbSysUserCommits

Suggested Presentation

counter

oraDbTablespaceTable

The oraDbTablespaceTable contains information on tablespaces within a current Oracle database instance. A tablespace is a logical portion of an Oracle database used to allocate storage for table and index data. In a production environment, tables can fill up as transaction activity mounts. Monitoring tablespace activity is important to avoid exhausting the tablespaces and causing spaces.

Each entry in the oraDbTablespaceTable represents a given tablespace within a current database instance. Each entry is indexed first by rdbmsDbIndex, and then oraDbTablespaceIndex; thus, there will normally be a one-to-many relationship between an rdbmsDbTable entry and corresponding oraDbTablespaceTable entries. (For details about rdbmsDbIndex, see Appendix D, "Interpreting Variables of the Public RDBMS MIB". The current value of each oraDbTablespaceTable variable is retrieved directly from a join of DBA_TABLESPACES, DBA_DATA_FILES, and DBA_FREE_SPACE selecting for the tablespace in question.

Table B–2 lists each of these variables and its corresponding object.

Table B–2 oraDbTablespaceTable Variables and Corresponding Object IDs.

Variable Name	Object ID
oraDbTablespaceIndex	1.3.6.1.4.1.111.4.1.2.1.1
oraDbTablespaceName	1.3.6.1.4.1.111.4.1.2.1.2

Table B–2 (Cont.) oraDbTablespaceTable Variables and Corresponding Object IDs.

Variable Name	Object ID
oraDbTablespaceSizeAllocated	1.3.6.1.4.1.111.4.1.2.1.3
oraDbTablespaceSizeUsed	1.3.6.1.4.1.111.4.1.2.1.4
oraDbTablespaceState	1.3.6.1.4.1.111.4.1.2.1.5
oraDbTablespaceLargestAvailableChunk	1.3.6.1.4.1.111.4.1.2.1.6

oraDbTablespaceIndex

Syntax

INTEGER (1..2147483647)

Max-Access

not-accessible

Explanation

A numeric index, unique among tablespaces within a single Oracle database instance. This index is assigned to these tablespaces based on the alphabetical order of their names. On most platforms, a hard file limit of 1023 database files exists, so this number is the practical upper limit. Although some administrators prefer a one-to-one relationship between tablespaces and database files, multiple files can comprise a tablespace.

Typical Range

1 - 1,000

Significance

very important

Related Variables

oraDbTablespaceName, rdbmsDbIndex

Suggested Presentation

not applicable

oraDbTablespaceName

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the name of this tablespace, as retrieved from the value for TABLESPACE-NAME in DBA_TABLESPACES.

Typical Range

0 - 30 characters

Significance

important

Related Variables

oraDbTablespaceIndex

Suggested Presentation

simple string

oraDbTablespaceSizeAllocated

Syntax

Integer32

Max-Access

read-only

Explanation

Indicates the amount of disk space (in kilobytes) allocated for this tablespace. This is the sum of the sizes of the data files associated with the tablespace.

Typical Range

0 - 10 gigabytes

Significance

important

Related Variables

oraDbTablespaceSizeUsed

Suggested Presentation

simple string

oraDbTablespaceSizeUsed

Syntax
Integer32

Max-Access
read-only

Explanation
Indicates the amount of disk space (in kilobytes) which is actually in use for storing data. This is the difference between the sum of the size of the data files associated with the tablespace and the sum of the size of the free spaces associated with the tablespace.

Typical Range
0 - value of oraDbTablespaceSizeAllocated
Significance

Very important

Significance
Very important

Related Variables
oraDbTablespaceSizeAllocated

Suggested Presentation
gauge

oraDbTablespaceState

Syntax
INTEGER (1) online (2) offline, or (3) invalid

Max-Access
read-only

Explanation
Indicates the current accessibility of this tablespace, as retrieved from the value for STATUS in DBA_TABLESPACES. If a tablespace is offline (2), then SQL statements cannot reference objects contained in the tablespace. An invalid (3) tablespace is one that has been dropped.

Typical Range
enumerated 1 - 3

Significance
very important

Related Variables

none

Suggested Presentation

simple string or icon

oraDbTablespaceLargestAvailableChunk

Syntax

Integer32

Max-Access

read-only

Explanation

Indicates the size (in kilobytes) of the largest contiguous set of free data blocks in the tablespace. This variable is retrieved directly from the value for NEXT_EXTENT in DBA_TABLES. If there is any object in the tablespace whose NEXT_EXTENT value is greater than the layout chunk, that object will be unable to expand. This might cause a major problem, or it might not matter.

Typical Range

Less than or equal to (oraDbTablespaceSizeAllocated minus oraDbTablespaceSizeUsed)

Significance

important

Related Variables

oraDbTablespaceSizeAllocated, oraDbTablespaceSizeUsed

Suggested Presentation

gauge

oraDbDataFileTable

oraDbDataFileTable contains information on the data files within a current database instance on the managed node. A data file denotes an area of disk allocated for database data. Monitoring data files is important for two reasons; first, to determine whether space in files is being exhausted, and second to determine O/S response time in accessing data on disk, especially on platforms where there is no other way to measure disk queue length.

Each entry in the oraDbDataFileTable represents a given data file within a current database instance on the node. Each entry is indexed first by rdbmsDbIndex, and then by oraDbDataFileIndex; thus, there will normally be a one-to-many relationship between an rdbmsDbTable entry and corresponding oraDbDataFileTable entries. (For details about rdbmsDbIndex, see Appendix D, "Interpreting Variables of the Public RDBMS MIB") The current value of each oraDbDataFileTable variable is retrieved directly from a corresponding field in either V$FILESTAT or DBA_DATA_FILES for that data file within that database instance.

Table B–3 lists each variable of the oraDbDataFileTable and its corresponding object ID.

Table B–3 oraDbDataFileTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbDataFileIndex	1.3.6.1.4.1.111.4.1.3.1.1
oraDbDataFileName	1.3.6.1.4.1.111.4.1.3.1.2
oraDbDataFileSizeAllocated	1.3.6.1.4.1.111.4.1.3.1.3
oraDbDataFileDiskReads	1.3.6.1.4.1.111.4.1.3.1.4
oraDbDataFileDiskWrites	1.3.6.1.4.1.111.4.1.3.1.5
oraDbDataFileDiskReadBlocks	1.3.6.1.4.1.111.4.1.3.1.6
oraDbDataFileDiskWrittenBlocks	1.3.6.1.4.1.111.4.1.3.1.7
oraDbDataFileDiskReadTimeTicks	1.3.6.1.4.1.111.4.1.3.1.8
oraDbDataFileDiskWriteTimeTicks	1.3.6.1.4.1.111.4.1.3.1.9

oraDbDataFileIndex

Syntax

INTEGER(1..2147483647)

Max-Access

not-accessible

Explanation

A numeric index, unique among data files associated with a single tablespace. The value of oraDbDataFileIndex is retrieved directly from the value of FILE_ID in DBA_DATA_FILES. For a small database, a tablespace typically contains 1 data file; for a large database (80 Gbytes), a tablespace typically contains 200 data files in a tablespace. The upper limit is hard coded as 1023.

Typical Range

Significance

very important

Related Variables

rdbmsDbIndex

Suggested Presentation

not applicable

oraDbDataFileName

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the fully-qualified name of this data file. The value is retrieved from the value of FILE_NAME in DBA_DATA_FILES.

Typical Range

10 - 100 characters; about 15 for raw partitions.

Significance

important

Related Variables

oraDbTablespaceName, because the file is a member of a tablespace.

Suggested Presentation

simple string

oraDbDataFileSizeAllocated

Syntax

Integer32

Max-Access

read-only

Explanation

Indicates the allocated size (in kilobytes) of this data file, as retrieved from the value of BYTES in DBA_DATA_FILES. Indicates how much space has been used so far. Monitoring this variable is very important, because running out of space can require taking the database down, depending on which table this file supports. Normally, however, adding another data file to the Tablespace solves the problem.

Typical Range

100 Mbytes - 2 Gbytes

Significance

very important

Related Variables

none

Suggested Presentation

simple string

oraDbDataFileDiskReads

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the total number of reads issued against this data file since database instance startup, as retrieved from the value of PHYRDS in V$FILESTAT.

Typical Range

10,000 - 100,000 or much more, depending on the life of the instance

Significance

less important

Related Variables

oraDbDataFileDiskReadBlocks, oraDbDataFileDiskReadTimeTicks

Suggested Presentation

counter

oraDbDataFileDiskWrites

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the total number of writes issued against this data file since database instance startup, as retrieved from the value of PHYWRTS in V$FILESTAT.

Typical Range

10,000 - 100,000 or much more, depending on the life of the instance

Significance

less important

Related Variables

oraDbDataFileDiskWriteBlocks, oraDbDataFileDiskWriteTimeTicks

Suggested Presentation

counter

oraDbDataFileDiskReadBlocks

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the total number of physical blocks read from this data file since database instance startup. The value is retrieved from the value of PHYBLKRD in V$FILESTAT. The number of bytes read can be determined by multiplying this value by the value of oraDbConfigDbBlockSize, typically 2Kbytes.

Typical Range

10,000 - 100,000 or much more, depending on the life of the instance

Significance

important

Related Variables

oraDbConfigDbBlockSize, oraDbDataFileDiskReads

Suggested Presentation

counter

oraDbDataFileDiskWrittenBlocks

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the total number of physical blocks written to this data file since database instance startup. The value is retrieved from the value of PHYBLKWRT in V$FILESTAT. The number of bytes written can be determined by multiplying this value by the value of the INIT.ORA parameter, DB_BLOCK_SIZE, typically 2Kbytes.

Typical Range

10,000 - 100,000 or much more, depending on the life of the instance

Significance

important

Related Variables

oraDbDataFileDiskWrites

Suggested Presentation

counter

oraDbDataFileDiskReadTimeTicks

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the time spent writing to this data file since database instance startup IF the INIT.ORA parameter TIMED_STATISTICS is TRUE. (The unit of measurement on a UNIX platform is time ticks of 10 ms.) If TIMED_STATISTICS is FALSE, then a value of 0 is returned. The value of oraDbDataFileDiskReadTimeTicks is retrieved from the value of READTIM in V$FILESTAT.

Because knowing how long it takes to perform I/O per block is the important statistic, knowing this aggregate measure is not very significant. If the average time to do reads is generally exceeding a couple of ticks, then an I/O sybsystem response time problem exists. Most high performance disk drives can easily locate any sector on the disk in 10 ms.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

oraDbDataFileDiskReads

Suggested Presentation

clock

oraDbDataFileDiskWriteTimeTicks

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the time spent writing to this data file since database instance startup IF the INIT.ORA parameter TIMED_STATISTICS is TRUE. (The unit of measurement on a UNIX platform is time ticks of 10 ms.) If TIMED_STATISTICS is FALSE, then a value of 0 is returned. The value of oraDbDataFileDiskWriteTimeTicks is retrieved from the value of WRITETIM in V$FILESTAT.

Because knowing how long it takes to perform I/O per block is the important statistic, knowing this aggregate measure is not very significant. If the average time to do writes is generally exceeding three ticks on non-mirrored data, then an I/O subsystem response time problem exists. Most high performance disk drives can easily perform a write in 15 ms.

Typical Range

Can wrap around the upper limit of 2147483647 in about three days of continuous instance operation for a busy database

Significance

less important

Related Variables

oraDbDataFileDiskWrites

Suggested Presentation

clock

oraDbLibraryCacheTable

The oraDbLibraryCacheTable contains variables measuring library cache activity since the most recent database instance startup. A library cache is a memory structure containing shared SQL and PL/SQL areas. Monitoring the library cache is important to determine whether it is necessary to resize Oracle's shared pool.

Each entry in the oraDbLibraryCacheTable represents a given library cache within a current database instance on the node. Each entry is indexed first by rdbmsDbIndex, and then by oraDbLibraryCacheIndex; thus, there will normally be a one-to-many relationship between an rdbmsDbTable entry and corresponding oraDbLibraryCacheTable entries. (For details about rdbmsDbIndex, see Appendix D, "Interpreting Variables of the Public RDBMS MIB") The current value of each oraDbLibraryCacheTable variable is retrieved directly from a corresponding field in the V$LIBRARYCACHE table for that library cache within that database instance.

Table B–4 lists each variable of the oraDbLibraryCacheTable and its corresponding object ID.

Table B–4 oraDbLibraryCacheTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbLibraryCacheIndex	1.3.6.1.4.1.111.4.1.4.1.1
oraDbLibraryCacheNameSpace	1.3.6.1.4.1.111.4.1.4.1.2
oraDbLibraryCacheGets	1.3.6.1.4.1.111.4.1.4.1.3
oraDbLibraryCacheGetHits	1.3.6.1.4.1.111.4.1.4.1.4
oraDbLibraryCachePins	1.3.6.1.4.1.111.4.1.4.1.5
oraDbLibraryCachePinHits	1.3.6.1.4.1.111.4.1.4.1.6
oraDbLibraryCacheReloads	1.3.6.1.4.1.111.4.1.4.1.7
oraDbLibraryCacheInvalidations	1.3.6.1.4.1.111.4.1.4.1.8

oraDbLibraryCacheIndex

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

A unique integer for each row of the oraDbLibraryCacheTable.

Typical Range

8 (only)

Significance

very important

Related Variables

oraDbLibraryCacheNameSpace

Suggested Presentation

not applicable

oraDbLibraryCacheNameSpace

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the namespace of the V$LIBRARYCACHE table to which this row is related. The value is retrieved from the value of NAMESPACE in V$LIBRARYCACHE.

Typical Range

0 - 30 characters

Significance

important

Related Variables

oraDbLibraryCacheIndex

Suggested Presentation

simple string

oraDbLibraryCacheGets

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the system requests handles to library objects in this namespace. The value of oraDbLibraryCacheGets is retrieved from the value of GETS in V$LIBRARYCACHE.

Typical Range

0 - 500,00 per day

Significance

important

Related Variables

oraDbLibraryCacheGetHits

Suggested Presentation

counter

oraDbLibraryCacheGetHits

Syntax
Counter32

Max-Access
read-only

Explanation
Indicates the number of times, since database instance startup, that the handles are already allocated in the cache. The value of oraDbLibraryCacheGetHits is retrieved from the value of GETHITS in V$LIBRARYCACHE.

Typical Range
0 - 4294967295

Significance
less important

Related Variables
oraDbLibraryCacheGets

Suggested Presentation
counter

oraDbLibraryCachePins

Syntax
Counter32

Max-Access
read-only

Explanation
Indicates the number of times, since database instance startup, that the system issues pin requests in order to access objects in the cache. The value of oraDbLibraryCachePins is retrieved from the value of PINS in V$LIBRARYCACHE.

oraDbLibraryCachePins is used to calculate the following database instance performance ratio: library cache miss ratio.

Typical Range
0 - 4294967295

Significance
less important

Related Variables

oraDbLibraryPinHits, oraDbLibraryCacheReloads

Suggested Presentation

counter

oraDbLibraryCachePinHits

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the objects the system is pinning are already allocated and initialized in the cache. The value of oraDbLibraryCachePinHits is retrieved from the value of PINHITS in V$LIBRARYCACHE.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbLibraryCachePins

Suggested Presentation

counter

oraDbLibraryCacheReloads

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that library objects have to be reinitialized and reloaded with data because they have been aged out or invalidated. The value of oraDbLibraryCacheReloads is retrieved from the value of RELOADS in V$LIBRARYCACHE.

oraDbLibraryCacheReloads is used to calculate the following database instance performance ratio: library cache miss ratio.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbLibraryCachePins

Suggested Presentation

counter

oraDbLibraryCacheInvalidations

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that non-persistent library objects (like shared SQL areas) have been invalidated. The value of oraDbLibraryCacheInvalidations is retrieved from the value of INVALIDATIONS in V$LIBRARYCACHE.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

none

Suggested Presentation

counter

oraDbLibraryCacheSumTable

The LibraryCacheSum table tracks for all library caches in a database instance as a whole the same information that the LibraryCache table tracks for the caches individually. It is primarily a convenience.

Table B–5 oraDBLibraryCacheSum Table Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbLibraryCacheSumGets	1.3.6.1.4.1.111.4.1.5.1.1
oraDbLibraryCacheSumGetHits	1.3.6.1.4.1.111.4.1.5.1.2
oraDbLibraryCacheSumPins	1.3.6.1.4.1.111.4.1.5.1.3
oraDbLibraryCacheSumPinHits	1.3.6.1.4.1.111.4.1.5.1.4
oraDbLibraryCacheSumReloads	1.3.6.1.4.1.111.4.1.5.1.5
oraDbLibraryCacheSumInvalidations	1.3.6.1.4.1.111.4.1.5.1.6

oraDbLibraryCacheSumGets

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the system requests handles to library objects in this namespace. The value of oraDbLibraryCacheSumGets is derived from the value of GETS in V$LIBRARYCACHE.

Typical Range

0 - 500,00 per day

Significance

important

Related Variables

oraDbLibraryCacheSumGetHits

Suggested Presentation

counter

oraDbLibraryCacheSumGetHits

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the handles are already allocated in the cache. The value of oraDbLibraryCacheSumGetHits is derived from the value of GETHITS in V$LIBRARYCACHE.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

oraDbLibraryCacheSumGets

Suggested Presentation

counter

oraDbLibraryCacheSumPins

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the system issues pin requests in order to access objects in the cache. The value of oraDbLibraryCacheSumPins is derived from the value of PINS in V$LIBRARYCACHE.

oraDbLibraryCacheSumPins is used to calculate the following database instance performance ratio: library cache miss ratio.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

oraDbLibraryPinHits, oraDbLibraryCacheSumReloads

Suggested Presentation

counter

oraDbLibraryCacheSumPinHits

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that the objects the system is pinning are already allocated and initialized in the cache. The value of oraDbLibraryCacheSumPinHits is derived from the value of PINHITS in V$LIBRARYCACHE.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbLibraryCacheSumPins

Suggested Presentation

counter

oraDbLibraryCacheSumReloads

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that library objects have to be reinitialized and reloaded with data because they have been aged out or invalidated. The value of oraDbLibraryCacheSumReloads is derived from the value of RELOADS in V$LIBRARYCACHE.

oraDbLibraryCacheSumReloads is used to calculate the following database instance performance ratio: library cache miss ratio.

Typical Range

0 - 4294967295

Significance

important

Related Variables

oraDbLibraryCacheSumPins

Suggested Presentation

counter

oraDbLibraryCacheSumInvalidations

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of times, since database instance startup, that non-persistent library objects (like shared SQL areas) have been invalidated. The value of oraDbLibraryCacheSumInvalidations is derived from the value of INVALIDATIONS in V$LIBRARYCACHE.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

none

Suggested Presentation

counter

oraDbSGATable

The oraDbSGATable contains storage allocation information on the current System Global Area (SGA). The SGA is a common database storage area for information that supports current sessions. As the values of its attributes are configured in the INIT.ORA file, they do not change while the database instance is running, thus, the SGA at this aggregate level is not as important to monitor as other database structures.

Each entry in the oraDbSGATable represents the SGA for a current database instance on the node. Each entry is indexed by rdbmsDbIndex. However, because rdbmsDbTable contains an entry for each installed database configured for SNMP support, whether or not it is actively open, there may or may not be a one-to-one correspondence between entries in the two tables at any point in time. (For details about rdbmsDbIndex, see Appendix D, "Interpreting Variables of the Public RDBMS MIB") The current value of each oraDbSGATable variable is retrieved directly from a corresponding field in the V$SGA table for that database instance.

Table B–6 lists each variable of the oraDbSGATable and its corresponding object ID.

Table B–6 oraDbSGATable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbSGAFixedSize	1.3.6.1.4.1.111.4.1.6.1.1
oraDbSGAVariableSize	1.3.6.1.4.1.111.4.1.6.1.2
oraDbSGADatabaseBuffers	1.3.6.1.4.1.111.4.1.6.1.3
oraDbSGARedoBuffers	1.3.6.1.4.1.111.4.1.6.1.4

oraDbSGAFixedSize

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The size in kilobytes of the fixed-size portion of the SGA as shown in V$SGASTAT.

Typical Range

1 - 2147483647

Significance

less important

Related Variables

oraDbSGAVariableSize

Suggested Presentation

as a simple integer value

oraDbSGAVariableSize

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
The size in kilobytes of the variable-size portion of the SGA as shown in V$SGASTAT.

Typical Range
1 - 2147483647

Significance
less important

Related Variables
oraDbSGAFixedSize

Suggested Presentation
as a simple integer value

oraDbSGADatabaseBuffers

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
The number of database buffers allocated in the SGA as shown in V$SGASTAT.

Typical Range
4 Kbytes - 2 Gbytes

Significance
less important

Related Variables
oraDbSGARedoBuffers

Suggested Presentation
as a simple integer value

oraDbSGARedoBuffers

Syntax
INTEGER (1..2147483647)

Max-Access
read/write

Explanation
The number of redo buffers allocated in the SGA as shown in V$SGASTAT.

Typical Range
1 - 2147483647

Significance
less important

Related Variables
oraDbSGADatabaseBuffers

Suggested Presentation
as a simple integer value

oraDbConfigTable

The oraDbConfigTable contains variables whose values are retrieved directly from the V$PARAMETER table. These parameters control systemwide resources and can impact the values of variables in the rest of this MIB. This table includes those initialization parameters that closely relate to the task of monitoring database instance performance. By assigning these parameters easy-to-use, fixed object IDs, it is easier to graph them together with the dynamic performance values they impact. For more information on these parameters, see the *Oracle Database Reference* and also the *Oracle Database Concepts* specific to your system.

Each entry in the oraDbConfigTable is indexed by rdbmsDbIndex. However, because rdbmsDbTable contains an entry for each installed database, whether or not it is actively open, there may or may not be a one-to-one correspondence between entries in the two tables at any point in time. (For details about rdbmsDbIndex, see Appendix D, "Interpreting Variables of the Public RDBMS MIB")

Table B–7 lists each variable of the oraDbConfigTable and its corresponding object ID.

Table B–7 oraDbConfigTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbConfigDbBlockBuffers	1.3.6.1.4.1.111.4.1.7.1.1
oraDbConfigDbBlockSize	1.3.6.1.4.1.111.4.1.7.1.3
oraDbConfigDbMultiBlockReadCount	1.3.6.1.4.1.111.4.1.7.1.5
oraDbConfigLogBuffer	1.3.6.1.4.1.111.4.1.7.1.11
oraDbConfigLogCheckpointInteval	1.3.6.1.4.1.111.4.1.7.1.12

Table B–7 (Cont.) oraDbConfigTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDbConfigLogCheckpointTimeout	1.3.6.1.4.1.111.4.1.7.1.13
oraDbConfigOpenCursors	1.3.6.1.4.1.111.4.1.7.1.19
oraDbConfigOpenLinks	1.3.6.1.4.1.111.4.1.7.1.20
oraDbConfigOptimizerMode	1.3.6.1.4.1.111.4.1.7.1.21
oraDbConfigProcesses	1.3.6.1.4.1.111.4.1.7.1.22
oraDbConfigSessions	1.3.6.1.4.1.111.4.1.7.1.24
oraDbConfigSharedPool	1.3.6.1.4.1.111.4.1.7.1.25
oraDbConfigSortAreaSize	1.3.6.1.4.1.111.4.1.7.1.26
oraDbConfigSortAreaRetainedSize	1.3.6.1.4.1.111.4.1.7.1.27
oraDbConfigTransactions	1.3.6.1.4.1.111.4.1.7.1.28
oraDbConfigTransactionsPerRollback	1.3.6.1.4.1.111.4.1.7.1.29

oraDbConfigDbBlockBuffers

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
The value of this variable is retrieved directly from the DB_BLOCK_BUFFERS parameter of the INIT.ORA file. Indicates the number of database blocks cached in memory of the SGA. This parameter is the most significant determinant of the SGA size and database instance performance. The size of each block can be determined using oraDbConfigDbBlockSize.

Typical Range
1 - 2147483647

Significance
very important

Related Variables
oraDbConfigDbBlockSize

Suggested Presentation
gauge or simple string

oraDbConfigDbBlockSize

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the DB_BLOCK_SIZE parameter of the INIT.ORA file. Indicates the size, in bytes, of database blocks for this database. This information has significance for a number of other variables.

Typical Range

1024 - 8192 (O/S dependent)

Significance

important

Related Variables

none

Suggested Presentation

gauge or simple string

oraDbConfigDbMultiBlockReadCount

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the DB_FILE_MULTIBLOCK_READ_COUNT parameter of the INIT.ORA file. Indicates the maximum number of blocks read in one I/O operation during a sequential scan. The default is a function of the DB_BLOCK_BUFFERS and PROCESSES parameters in the INIT.ORA file.

Typical Range

Values in the range of 4 to 32 are reasonable. Actual maximums vary by operating system

Significance

important

Related Variables

oraDbConfigDbBlockBuffers, oraDbConfigDbProcesses

Suggested Presentation

gauge or simple string

oraDbConfigLogBuffer

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the LOG_BUFFER parameter of the INIT.ORA file. Indicates the number of bytes allocated to the redo log buffer in the SGA. In a busy system, a value of 65536 or higher would not be unreasonable.

Typical Range

1,000 - 100,000

Significance

important

Related Variables

none

Suggested Presentation

gauge or simple string

oraDbConfigLogCheckpointInterval

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the LOG_CHECKPOINT_ INTERVAL parameter of the INIT.ORA file. Indicates the amount of time (in seconds) to pass before another checkpoint occurs. A value of 0 disables time-based checkpoints.

Typical Range

1 - 2147483647

Significance

important

Related Variables

oraDbConfigLogCheckpointTimeout

Suggested Presentation

clock or simple string

oraDbConfigLogCheckpointTimeout

Syntax

INTEGER (0..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the LOG_CHECKPOINT_ TIMEOUT parameter of the INIT.ORA file. The amount of time (in seconds) to pass between checkpoints. A value of 0 disables time-based checkpoints.

Typical Range

0 - 2147483647

Significance

important

Related Variables

oraDbConfigLogCheckpointInterval

Suggested Presentation

clock or simple string

oraDbConfigOpenCursors

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the OPEN_CURSORS parameter of the INIT.ORA file. Indicates the maximum number of open cursors (context areas) a single-user process can have at once.

Typical Range

1 - 2147483647

Significance

important

Related Variables

none

Suggested Presentation

gauge or simple string

oraDbConfigOpenLinks

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the OPEN_LINKS parameter of the INIT.ORA file. Indicates the maximum number of concurrent open connections to remote database instances per user process. This parameter refers only to connections used for distributed transactions; direct connections to a remote database instance specified as an application connects are not counted. If set to 0, then no distributed transactions are allowed.

Typical Range

0 - 255

Significance

important

Related Variables

none

Suggested Presentation

gauge or simple string

oraDbConfigOptimizerMode

Syntax

DisplayString

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the OPTIMIZER_MODE parameter of the INIT.ORA file. When set to RULE, this parameter causes rule-based optimization to be used, unless hints are specified in the query. When set to COST, the optimizer uses a cost-based approach for an SQL statement, if there are statistics in the dictionary for at least one table accessed in the statement. (Otherwise, the rule-based approach is used.)

Typical Range

RULE/COST

Significance

important

Related Variables

oraDbConfigSessions

Suggested Presentation

icon or simple string

oraDbConfigProcesses

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the PROCESSES parameter of the INIT.ORA file. For a multiple-process operation, this parameter specifies the maximum number of user processes that can simultaneously connect to an Oracle server. The PROCESSES parameter is used to derive the SESSIONS INIT.ORA parameter. The default is 50.

Typical Range

6 - O/S dependent

Significance

important

Related Variables

oraDbConfigSessions, oraDbConfigTransactions

Suggested Presentation

gauge or simple string

Suggested Presentation

gauge or simple string

oraDbConfigSessions

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the SESSIONS parameter of the INIT.ORA file. Indicates the total number of user and system sessions. The default for parameter is derived as the value of the INIT.ORA PROCESSES parameter multiplied by 1.1.

Typical Range

1 - 2147483647

Significance

important

Related Variables

oraDbConfigProcesses

Suggested Presentation

gauge or simple string

oraDbConfigSharedPool

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the SHARED_POOL_SIZE parameter of the INIT.ORA file. Indicates the size (in bytes) of the shared pool. The shared pool contains shared cursors and stored procedures. The default is 3.5 Mbytes.

Typical Range

300 Kbytes - O/S dependent

Significance

important

Related Variables

none

Suggested Presentation

gauge or simple string

oraDbConfigSortAreaSize

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the SORT_AREA_SIZE parameter of the INIT.ORA file. Indicates the maximum amount of PGA memory to use for an external (disk) sort (in bytes). The default (O/S dependent) is adequate for most DB operations.

Typical Range

1 - 2147483647

Significance

none

Related Variables

oraDbConfigRetainedSize, oraDbConfigSortSpaceMapSize

Suggested Presentation

gauge or simple string

oraDbConfigSortAreaRetainedSize

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
The value of this variable is retrieved directly from the SORT_AREA_RETAINED_ SIZE parameter of the INIT.ORA file. Indicates the maximum amount of session memory (in bytes) that will be used for an in-memory sort. The default value is equal to that assigned for the SORT_AREA_SIZE INIT.ORA parameter.

Typical Range
0..value of oraDbConfigSortAreaSize

Significance
important

Related Variables
oraConfigSortAreaSize

Suggested Presentation
gauge or simple string

oraDbConfigTransactions

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
The value of this variable is retrieved directly from the TRANSACTIONS parameter of the INIT.ORA file. Indicates the maximum number of concurrent transactions. The default is the value of the PROCESSES INIT.ORA parameter multiplied by 1.1.

Typical Range
1 - 2147483647

Significance
important

Related Variables
oraDbConfigProcesses

Suggested Presentation

gauge or simple string

oraDbConfigTransactionsPerRollback

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

The value of this variable is retrieved directly from the TRANSACTIONS_PER_ ROLLBACK_SEGMENT parameter of the INIT.ORA file. Indicates the number of concurrent transactions allowed per rollback segment. The default is 30.

Typical Range

1..O/S dependent

Significance

important

Related Variables

oraDbConfigTransactions

Suggested Presentation

gauge or simple string

C

Interpreting Variables of the Oracle Network Listener MIB

This appendix provides information to help you interpret individual variables in the private Oracle Network Listener MIB. Specifically, it includes sections covering MIB variables in the following tables:

- oraListenerTable
- oraSIDTable
- oraDedicatedSrvTable
- oraDispatcherTable
- oraPrespawnedSrvTable
- oraListenAddressTable
- oraListenerTraps

For more information about the Oracle network listener, see *Oracle Database Net Services Administrator's Guide*.

oraListenerTable

The oraListenerTable contains information about the generic network listeners installed on a managed node. A network listener is a server process that listens for connection requests for one or more databases on one or more network protocols. Monitoring a listener is important in order to ensure adequate database access.

Each entry in the oraListenerTable represents the network listener for a current database instance on the node. Each entry is indexed by oraListenerIndex. The current value of each oraListenerTable variable is retrieved from either configuration files or calculated from an internal buffer of the managed listener.

Table C–1 lists the variables included in the oraListenerTable.

Table C–1 oraListenerTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraListenerIndex	1.3.6.1.4.1.111.5.1.1.1.1
oraListenerName	1.3.6.1.4.1.111.5.1.1.1.2
oraListenerVersion	1.3.6.1.4.1.111.5.1.1.1.3
oraListenerStartDate	1.3.6.1.4.1.111.5.1.1.1.4
oraListenerUptime	1.3.6.1.4.1.111.5.1.1.1.5

Table C–1 (Cont.) oraListenerTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraListenerTraceLevel	1.3.6.1.4.1.111.5.1.1.1.6
oraListenerSecurityLevel	1.3.6.1.4.1.111.5.1.1.1.7
oraListenerParameterFile	1.3.6.1.4.1.111.5.1.1.1.8
oraListenerLogFile	1.3.6.1.4.1.111.5.1.1.1.9
oraListenerTraceFile	1.3.6.1.4.1.111.5.1.1.1.10
oraListenerState	1.3.6.1.4.1.111.5.1.1.1.11
oraListenerNumberOfServices	1.3.6.1.4.1.111.5.1.1.1.12
oraListenerContact	1.3.6.1.4.1.111.5.1.1.1.13

A description of each of these variables follows.

oraListenerIndex

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
A numeric index, unique among all generic listeners on this managed node, as retrieved from the snmp.index.*listener-name* parameter of the SNMP_RW.ORA configuration file. oraListenerIndex also serves as the primary index for the oraListenAddressTable.

Typical Range
varies, depending on the value set in the snmp.index.*listener-name* field of the SNMP_RW.ORA configuration file

Significance
very important

Related Variables
oraListenAddress

Suggested Presentation
not applicable

oraListenerName

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the name of this generic listener, as retrieved from the LISTENER.ORA configuration file.

Typical Range

1 - 255 characters

Significance

very important

Related Variables

none

Suggested Presentation

icon or simple string

oraListenerVersion

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the version number of this generic listener, in vendor-specific format. This value is retrieved from an internal listener table.

Typical Range

30 - 50 characters (physical limit is 255 characters)

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraListenerStartDate

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the date and time at which this generic listener was started, as retrieved from an internal listener table. The format is date-month-year hours:minutes:seconds.

Typical Range

15 - 20 characters (physical limit is 255 characters)

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraListenerUptime

Syntax

TimeTicks

Max-Access

read-only

Explanation

Indicates the value of sysUpTime at the time the generic listener was started. This value is derived from the master agent, which counts the time in hundredths of a second since the master agent was started. The value of this variable should remain as a constant across the Listener's life span. A change in this variable indicates that the counter has been refreshed.

Typical Range

0 - 4294967295

Significance

important

Related Variables

none

Suggested Presentation

clock

oraListenerTraceLevel

Syntax

INTEGER (1 - 17)

Max-Access

read/write

Explanation

Indicates the level at which a Listener should be traced. A value of 4 means that tracing at the USER level is turned on; and 6 means that tracing at the ADMIN level is turned on. A value of 17 means that tracing is turned off.

Typical Range

enumerated 1 - 17

Significance

important

Related Variables

oraNavigatorTraceFile

Suggested Presentation

as an icon, or as a simple string of characters, OFF, USER or ADMIN

oraListenerSecurityLevel

Syntax

INTEGER (1) on, or (2) off.

Max-Access

read-only

Explanation

Indicates whether or not the generic listener is password-protected, as determined from the PASSWORDS_listener_name parameter of the LISTENER.ORA configuration file. If security is ON (1), the password is required to perform certain DBA tasks against the Listener using the Listener control utility.

Typical Range

enumerated 1 or 2

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraListenerParameterFile

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the name of the Listener configuration file on the managed node, as retrieved from an internal buffer of the managed listener. Default is $ORACLE_ HOME/network/admin/listener_name.ORA

Typical Range

40 - 70 characters (physical limit is 255 characters)

Significance

important

Related Variables

none

Suggested Presentation

simple string

oraListenerLogFile

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the directory for the log file that is automatically generated for listener events, as retrieved from the LOG_DIRECTORY parameter of the LISTENER.ORA configuration file. Default is operating system specific: on UNIX, for example, it is $ORACLE_HOME/network/log/listener_name.log.

Typical Range

20 - 70 characters (physical limit is 255 characters)

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraListenerTraceFile

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the name of the file to which trace information is written, as determined from the TRACE_FILE parameter of the LISTENER.ORA configuration file. Default is $ORACLE_HOME/network/trace/listener_name.trc

Typical Range

20 - 70 characters (physical limit is 255 characters)

Significance

less important

Related Variables

oraListenerTraceLevel

Suggested Presentation

simple string

oraListenerState

Syntax
INTEGER (1) up, or (2) down

Max-Access
read-only

Explanation
Indicates the current state of the generic listener, as retrieved from an internal table. The DOWN value (2) indicates that the generic listener is not available for any use, but is known to the agent. The UP value (1) indicates that the generic listener is running and available for generic use. This value would be inspected by an agent on receipt of an oraListenerStateChange trap.

Typical Range
enumerated 1 or 2

Significance
important

Related Variables
oraListenerStateChange

Suggested Presentation
simple string or icon

oraListenerNumberOfServices

Syntax
INTEGER (0..2147483647)

Max-Access
read-only

Explanation
Indicates the number of different types of services, dedicated server, dispatcher, or prespawned shadow, configured with the listener:

Typical Range
1 - 25, depending on the configuration parameters

Significance
less important

Related Variables
none

Suggested Presentation

simple string

oraListenerContact

Syntax

DisplayString

Max-Access

read/write

Explanation

Free-form text providing contact information for this managed listener, including name, telephone number, e-mail address, for example. This value is retrieved from the snmp.contact.*listener_name* parameter of the SNMP_RW.ORA configuration file.

Range

0 - 255 characters

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraSIDTable

The oraSIDTable contains information about the status of connections to a current database instance. This information is important to monitor in order to ensure that clients seeking access to a database have that opportunity.

Each entry in the oraSIDTable represents a current database instance for whom the listener listens. Each entry is indexed by oraSIDName.

Table C–2 lists the variables included in the oraSIDTable.

Table C–2 oraSIDTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraSIDListenerIndex	1.3.6.1.4.1.111.5.1.5.1.1
oraSIDName	1.3.6.1.4.1.111.5.1.5.1.2
oraSIDCurrentConnectedClients	1.3.6.1.4.1.111.5.1.5.1.3
oraSIDReservedConnections	1.3.6.1.4.1.111.5.1.5.1.4

A description of each of these variables follows.

oraSIDListenerIndex

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

A numeric index, unique among all of the generic listeners on the managed node, that associates a listener with a database it listens on. This value is retrieved from the snmp.index.*listener_name* parameter of the SNMP_RW.ORA configuration file.

Typical Range

varies, depending on the value set in the snmp.index.*listener-name* field of the SNMP_RW.ORA configuration file

Significance

very important

Related Variables

oraListenerIndex

Suggested Presentation

not applicable

oraSIDName

Syntax

DisplayString

Max-Access

read-only

Explanation

A unique name identifying a database instance. This SID is in the definition of the listener, as retrieved from the SID_NAME parameter of the LISTENER.ORA configuration file. oraSIDName serves as the primary index in the oraDedicatedSrvTable, oraDispatcherTable, and oraPrespawnedSrvTable; thus, an entry for any dispatcher, dedicated server or prespawned server which listens for the database instance identified by oraSIDName can be found.

Typical Range

1 - 25 characters (physical limit is 255 characters)

Significance

very important

Related Variables

oraDedicatedSrvIndex, oraDispatcherIndex, oraPrespawnedSrvIndex

Suggested Presentation

simple string or icon

oraSIDCurrentConnectedClients

Syntax

Gauge32

Max-Access

read-only

Explanation

Indicates the total number of clients currently connected through prespawned shadows and dispatchers to this database instance, as retrieved from an internal Listener table. When the value of oraSIDCurrentConnectedClients is approaching the value of oraSIDReservedConnections, the number of connection requests to this database instance is reaching the threshold (the reserved number of connections for clients on this database instance).

When the value of oraSIDCurrentConnectedClients equals the value of oraSIDReservedConnections, the incoming connection requests are redirected to dedicated servers. This may not be desirable, as a dedicated server requires more system resources than does a dispatcher or prespawned shadow.

Typical Range

0 - value of oraSIDReservedConnections

Significance

very important

Related Variables

oraDispatcherCurrentConnections, oraPrespawnedSrvCurrentConnections

Suggested Presentation

gauge

oraSIDReservedConnections

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the total number of reserved connections for clients through prespawned shadows and dispatchers to this database instance.

Typical Range

0 - 1,000, depending on the number of dispatcher and prespawned shadow processes for the database

Significance

very important

Related Variables

oraDispatcherMaximumConnections, oraSIDCurrentConnectedClients

Suggested Presentation

gauge

oraDedicatedSrvTable

The oraDedicatedSrvTable contains information about the dedicated servers provided by the managed listener. A dedicated server is a process that executes Oracle server code on behalf of a particular database user process. Monitoring a dedicated server is important in order to ensure that clients seeking access to a database have that opportunity.

Each entry in the oraDedicatedSrvTable represents a dedicated server associated with a current database instance on the managed node. Each entry is indexed first by oraSIDName, and then by oraDedicatedSrvIndex; thus, there can be a one-to-many relationship between a current database instance and the dedicated servers associated with it. By indexing first by oraSIDName, all dedicated servers associated with a given SID can be found using SNMP traversal operations (get-next and get-bulk). The current value of each oraDedicatedSrvTable variable is retrieved from internal listener buffers.

Table C–3 lists the variables included in the oraDedicatedSrvTable.

Table C–3 oraDedicatedSrvTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDedicatedSrvIndex	1.3.6.1.4.1.111.5.1.2.1.1
oraDedicatedSrvEstablishedConnections	1.3.6.1.4.1.111.5.1.2.1.2
oraDedicatedSrvRejectedConnections	1.3.6.1.4.1.111.5.1.2.1.3

A description of each of these variables follows.

oraDedicatedSrvIndex

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

A numeric index, unique among all the dedicated servers, as retrieved from an internal listener table.

Typical Range

1 - 5, depending on the number of database instances for whom this listener listens

Significance

very important

Related Variables

oraSIDName

Suggested Presentation

not applicable

oraDedicatedSrvEstablishedConnections

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates how many incoming connection requests this dedicated server has accepted since startup of this managed listener, as retrieved from an internal listener table.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

oraDedicatedSrvRejectedConnections, oraDispatcherRejectedConnections, oraPrespawnedSrvRejectedConnections

Suggested Presentation

counter

oraDedicatedSrvRejectedConnections

Syntax
Counter32

Max-Access
read-only

Explanation
Indicates how many incoming connection requests this dedicated server has rejected since startup of this managed listener, as retrieved from an internal listener table.

Typical Range
0 - 4294967295

Significance
very important

Related Variables
oraDedicatedSrvEstablishedConnections

Suggested Presentation
counter

oraDispatcherTable

The oraDispatcherTable contains information on the dispatchers provided by the managed listener. A dispatcher is an optional background process present only when a multi-threaded server configuration is used. Monitoring a dispatcher is important in order to ensure that clients seeking access to a database have that opportunity.

Each entry in the oraDispatcherTable represents a dispatcher associated with a current database instance. Each entry is indexed first by oraSIDName, and then by oraDispatcherIndex; thus, there can be a one-to-many relationship between a database instance and the dispatcher processes serving it. Using this indexing method, all dispatchers associated with a given SID can be found by SNMP traversal operations (get-next and get-bulk). The current value of most oraDispatcherTable variables is retrieved from internal listener tables.

Table C–4 lists the variables included in the oraDispatcherTable.

Table C–4 oraDispatcherTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDispatcherIndex	1.3.6.1.4.1.111.5.1.3.1.1
oraDispatcherEstablishedConnections	1.3.6.1.4.1.111.5.1.3.1.2
oraDispatcherRejectedConnections	1.3.6.1.4.1.111.5.1.3.1.3
oraDispatcherCurrentConnections	1.3.6.1.4.1.111.5.1.3.1.4
oraDispatcherMaximumConnections	1.3.6.1.4.1.111.5.1.3.1.5
oraDispatcherState	1.3.6.1.4.1.111.5.1.3.1.6

Table C–4 (Cont.) oraDispatcherTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraDispatcherProtocolInfo	1.3.6.1.4.1.111.5.1.3.1.7

A description of each of these variables follows.

OraDispatcherIndex

Syntax
INTEGER (1..2147483647)

Max-Access
read-only

Explanation
A numeric index, unique among all the dispatchers, as retrieved from an internal listener buffer.

Typical Range
1 - 5, depending on the number of dispatchers configured for the database instance

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraDispatcherEstablishedConnections

Syntax
Counter32

Max-Access
read-only

Explanation
Indicates how many incoming connection requests this dispatcher has accepted since startup of this managed listener, as retrieved from an internal listener buffer.

When the value of oraDispatcherEstablishedConnections approaches the value of oraDispatcherMaximumConnections, this dispatcher is reaching the threshold (the greatest number of connection requests that can be accepted).

Typical Range

0 - value of oraDispatcherMaximumConnections

Significance

very important

Related Variables

oraDedicatedSrvEstablishedConnections, oraDispatcherMaximumConnections, oraDispatcherRejectedConnections, oraPrespawnedSrvEstablishedConnections

Suggested Presentation

counter

oraDispatcherRejectedConnections

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates how many incoming connection requests this dispatcher has rejected since startup of this managed listener, as retrieved from an internal listener table.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

oraDedicatedSrvEstablishedConnections, oraDispatcherEstablishedConnections, oraDispatcherState, oraPrespawnedSrvEstablishedConnections

Suggested Presentation

counter

oraDispatcherCurrentConnections

Syntax

Gauge32

Max-Access

read-only

Explanation

Indicates the number of connection requests this dispatcher is currently redirecting, as retrieved from an internal listener table. Monitoring oraDispatcherCurrentConnections and oraDispatcherMaximumConnections together indicates the load/activity of the dispatcher.

Typical Range

0 - oraDispatcherMaximumConnections

Significance

very important

Related Variables

oraSIDCurrentConnectedClients, oraDispatcherMaximumConnections

Suggested Presentation

gauge

oraDispatcherMaximumConnections

Syntax

INTEGER (1..21474836647)

Max-Access

read-only

Explanation

Indicates the greatest number of connection requests that this dispatcher can accept, as retrieved from an internal listener table.

Typical Range

0 - value of oraSIDReservedConnections

Significance

very important

Related Variables

oraDispatcherCurrentConnections, oraSIDReservedConnections

Suggested Presentation

gauge

oraDispatcherState

Syntax

INTEGER (1) blocked, or (2) ready.

Max-Access

read-only

Explanation

Indicates the current state of this dispatcher, as retrieved from an internal listener table. The BLOCKED value (1) indicates that the dispatcher is temporarily not accepting redirected connection requests. The READY value (2) indicates that this dispatcher is ready to accept any redirected connection requests.

Typical Range

enumerated 1 or 2

Significance

very important

Related Variables

oraDispatcherRejectedConnections

Suggested Presentation

icon or simple string

oraDispatcherProtocolInfo

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the protocol for which this dispatcher listens, as well as protocol-specific information.

Typical Range

1 - 255 characters

Significance

important

Related Variables

none

Suggested Presentation

simple string or icon

oraPrespawnedSrvTable

The oraPrespawnedSrvTable contains information on the prespawned servers provided by the managed listener associated with the current database instance. A prespawned server is an optional shadow process created when the listener is started, and then available for making connections to the database. Prespawned servers reduce connect time by eliminating the need to spawn a shadow for each new connect request. Monitoring a prespawned server is important in order to ensure that clients seeking access to a database have that opportunity.

Each entry in the oraPrespawnedSrvTable represents a prespawned server associated with a current database instance. Each entry is indexed first by oraSIDName, and then by oraPrespawnedSrvIndex; thus, there can be a one-to-many relationship between a database instance and the prespawned servers serving it. Using this indexing method, all prespawned servers associated with a given SID can be found by SNMP traversal operations (get-next and get-bulk). The current value of most oraPrespawnedSrvTable variables is retrieved from internal listener buffers.

Table C–5 lists the variables included in the oraPrespawnedSrvTable.

Table C–5 oraPrespawnedSrvTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraPrespawnedSrvIndex	1.3.6.1.4.1.111.5.1.4.1.1
oraPrespawnedSrvEstablishedConnections	1.3.6.1.4.1.111.5.1.4.1.2
oraPrespawnedSrvRejectedConnections	1.3.6.1.4.1.111.5.1.4.1.3
oraPrespawnedSrvCurrentConnections	1.3.6.1.4.1.111.5.1.4.1.4
oraPrespawnedSrvMaximumConnections	1.3.6.1.4.1.111.5.1.4.1.5
oraPrespawnedSrvState	1.3.6.1.4.1.111.5.1.4.1.6
oraPrespawnedSrvProtocolInfo	1.3.6.1.4.1.111.5.1.4.1.7
oraPrespawnedSrvProcessorID	1.3.6.1.4.1.111.5.1.4.1.8

A description of each of these variables follows.

oraPrespawnedSrvIndex

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

A numeric index, unique among the prespawned servers configured for this managed listener, as retrieved from an internal listener table.

Typical Range

1 - 1,000, depending on the number of prespawned shadows configured

Significance

very important

Related Variables

none

Suggested Presentation

not applicable

oraPrespawnedSrvEstablishedConnections

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates how many incoming connection requests have been accepted by the prespawned server since its startup, as retrieved from an internal listener table.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

none

Suggested Presentation

counter

oraPrespawnedSrvRejectedConnections

Syntax

Counter32

Max-Access

read-only

Explanation

Indicates the number of connection requests this prespawned server has rejected since its startup, as retrieved from an internal listener table. If the value of oraPrespawnedSrvRejectedConnections is creeping up, oraPrespawnedState should be monitored.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

oraDedicatedSrvEstablishedConnections, oraPrespawnedSrvEstablishedConnections, oraPrespawnedSrvState

Suggested Presentation

counter

oraPrespawnedSrvCurrentConnections

Syntax

Gauge32

Max-Access

read-only

Explanation

Indicates how many incoming connection requests currently being redirected by the prespawned server since startup of this managed listener, as retrieved from an internal listener table.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

oraPrespawnedSrvMaximumConnections

Suggested Presentation

gauge

oraPrespawnedSrvMaximumConnections

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

Indicates the greatest number of connection requests that this prespawned server can accept. The Listener updates this value in real time.

Typical Range

0 - 1,000, depending on the configuration

Significance

very important

Related Variables

oraPrespawnedSrvCurrentConnections

Suggested Presentation

gauge

oraPrespawnedSrvState

Syntax

INTEGER (1) blocked, or (2) ready.

Max-Access

read-only

Explanation

Indicates the current state of the prespawned server, as retrieved from an internal listener table. The BLOCKED value (1) indicates that the prespawned server is temporarily not accepting redirected connection requests. The READY option (2) indicates that the prespawned server is ready to accept any redirected connection requests.

Typical Range

enumerated 1 or 2

Significance

very important

Related Variables

oraPrespawnedSrvRejectedConnection

Suggested Presentation

alarm or icon

oraPrespawnedSrvProtocolInfo

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the protocol on which this prespawned server listens, as well as protocol-specific information. This information is retrieved from an internal Listener table.

Typical Range

1 - 255 characters

Significance

important

Related Variables

none

Suggested Presentation

simple string or icon

oraPrespawnedSrvProcessorID

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates the process ID of the prespawned server, as retrieved from an internal listener table.

Typical Range

1 - 10, depending on the O/S (physical limit is 255 characters)

Significance

less important

Related Variables

none

Suggested Presentation

simple string

oraListenAddressTable

The oraListenAddressTable contains the addresses on which the listener is listening. These addresses are important to monitor when a Listener cannot come up during startup; these addresses can then be examined for the proper format.

Each entry in the oraListenAddressTable represents an address on which a given active listener is listening. Each entry is indexed first by oraListenerIndex, and then by oraListenAddressIndex; thus, there can be a one-to-many relationship between a listener and the addresses on which it is listening.

Table C–6 lists the variables included in the oraListenAddressTable.

Table C–6 oraListenAddressTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraListenAddressIndex	1.3.6.1.4.1.111.5.1.6.1.1
oraListenAddress	1.3.6.1.4.1.111.5.1.6.1.2

A description of each of these variables follows.

oraListenAddressIndex

Syntax

INTEGER (1..2147483647)

Max-Access

read-only

Explanation

A numeric index, unique among all of the addresses on which the listener is listening, as retrieved from an internal listener table.

Typical Range

1 - 5, depending on configuration

Significance

important

Related Variables

oraListenAddress, oraListenIndex

Suggested Presentation

not applicable

oraListenAddress

Syntax

DisplayString

Max-Access

read-only

Explanation

Indicates an address on which this listener is listening, as retrieved from the ADDRESS section of the LISTENER.ORA configuration file.

Typical Range

1 - 255 characters

Significance

important

Related Variables

oraListenerIndex

Suggested Presentation

simple string

oraListenerTraps

Table C–7 lists the oraListenerTraps variable.

Table C–7 oraListenerTraps Variable and Corresponding Object ID

Variable Name	Object ID
oraListenerStateChange	1.3.6.1.4.1.111.5.2.1.1.1

A description of this variable follows.

oraListenerStateChange

Objects

{oraListenerState}

Explanation

Indicates that one of the listeners managed by this agent has changed its listenerState in a way that makes it less accessible for use.

Typical Range

2 (only)

Significance

very important

Related Variables

oraListenerState

Suggested Presentation

alarm

D

Interpreting Variables of the Public RDBMS MIB

This appendix provides information to help you interpret individual variables in the Oracle implementation of the public RDBMS MIB. Note that this appendix emphasizes Oracle's implementation. For information concerning other implementation alternatives for these variables, refer to MIB itself.

This appendix includes sections covering public RDBMS MIB variables that Oracle implements in the following tables

- rdbmsDbTable
- rdbmsDbInfoTable
- rdbmsSrvTable
- rdbmsSrvInfoTable
- rdbmsSrvParamTable
- rdbmsRelTable
- rdbmsTraps

Note that Oracle does not implement the following tables of the public RDBMS MIB:

- rdbmsDbParamTable
- rdbmsDbLimitedResourceTable
- rdbmsSrvLimitedResourceTable

Attempts to access any of the variables that Oracle is not implementing results in noSuchName (SNMPv1) or noSuchInstance (SNMPv2) being returned.

rdbmsDbTable

The rdbmsDbTable contains general information on each database installed on the managed node. Any installed database that has been configured for SNMP support, whether or not it is actively opened, is represented by a row in the rdbmsDbTable.

Each entry in the rdbmsDbTable represents a given database installed on the node. Each entry is indexed by rdbmsDbIndex. The current values of the rdbmsDbTable variables are retrieved from a variety of sources on the managed node.

Oracle implements all rdbmsDbTable variables; Table D–1 lists each of them, together with its corresponding object ID.

Table D–1 rdbmsDbTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsDbIndex	1.3.6.1.2.1.39.1.1.1.1
rdbmsDbPrivateMIBOID	1.3.6.1.2.1.39.1.1.1.2
rdbmsDbVendorName	1.3.6.1.2.1.39.1.1.1.3
rdbmsDbName	1.3.6.1.2.1.39.1.1.1.4
rdbmsDbContact	1.3.6.1.2.1.39.1.1.1.5

rdbmsDbIndex

Syntax

INTEGER (1 - 2147483647)

Max-Access

not-accessible

Explanation

A numeric index, unique among all of the vendors' databases installed on this managed node. This value is a surrogate for the conceptually unique key, which is {vendorOID, databasename}. In the Oracle implementation, the value of this variable when not running Oracle Real Application Clusters is the value of rdbmsSrvIndex for this installed database server.

Typical Range

1 - 10

Significance

very important

Related Variables

rdbmsSrvIndex

Suggested Presentation

not applicable

rdbmsDbPrivateMIBOID

Syntax

OBJECT IDENTIFIER

Max-Access

read-only

Explanation

Displays the authoritative identification for the private MIB of this installed database. In the Oracle implementation, the value of rdbmsDbVendorOID is hardcoded as the root of Oracle's OID subtree (1.3.6.1.4.1.111).

Typical Range

1.3.6.1.4.1.111 (only)

Significance

very important

Related Variables

rdbmsSrvVendorOID

Suggested Presentation

simple string

rdbmsDbVendorName

Syntax

DisplayString

Max-Access

read-only

Explanation

Displays the name of the vendor whose RDBMS manages this installed database, for informational purposes. In the Oracle implementation, the value of this variable is hardcoded in the subagent as "Oracle."

Typical Range

6 characters (only)

Significance

important

Related Variables

rdbmsSrvVendorName

Suggested Presentation

icon or simple string

rdbmsDbName

Syntax

DisplayString

Max-Access

read-only

Explanation

Displays the name of this installed database, in product-specific format. In the Oracle implementation, the value of rdbmsDbName is retrieved from snmp.visibleservices parameter of the SNMP_RO.ORA configuration file. This name is globally unique and includes domain information.

Typical Range

5 - 25 characters

Significance

important

Related Variables

rdbmsSrvName

Suggested Presentation

icon or simple string

rdbmsDbContact

Syntax

DisplayString

Max-Access

read/write

Explanation

Displays the name of the contact person for this installed database, together with information indicating how to contact this person. In the Oracle implementation, the value of this variable is retrieved from the SNMP_RW.ORA configuration file

Typical Range

2 - 255 characters

Significance

important

Related Variables

rdbmsSrvContact

Suggested Presentation

simple string

rdbmsDbInfoTable

The rdbmsDbInfoTable contains additional general information about actively opened database instances on the managed node. If an installed database has been configured for SNMP support but is not actively opened, then attempts to access corresponding instances in this table may result in either noSuchName (SNMPv1), or noSuchInstance (SNMPv2). "Actively opened" is determined by the values of all the rdbmsRelState entries for this database in the rdbmsRelTable.

Each entry in the rdbmsDbInfoTable represents a particular actively opened database instance on the node. Each entry is indexed by rdbmsDbIndex. However, because rdbmsDbTable contains an entry for each database installed on that node, whether or not it is actively open, there may or may not be a one-to-one correspondence between entries in the two tables at any point in time. The current value of the rdbmsDbInfoTable variable are retrieved from a variety of sources on the managed node.

Table D–2 lists each rdbmsDbInfoTable variable that Oracle is implementing, together with its corresponding object ID.

Table D–2 rdbmsDbInfoTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsDbInfoProductName	1.3.6.1.2.1.39.1.2.1.1
rdbmsDbInfoVersion	1.3.6.1.2.1.39.1.2.1.2
rdbmsDbInfoSizeUnits	1.3.6.1.2.1.39.1.2.1.3
rdbmsDbInfoSizeAllocated	1.3.6.1.2.1.39.1.2.1.4
rdbmsDbInfoSizeUsed	1.3.6.1.2.1.39.1.2.1.5
rdbmsDbInfoLastBackup*	1.3.6.1.2.1.39.1.2.1.6

* Variables marked with an asterisk are not supported.

Attempts to access these variables may result in errors.

Note that Oracle is not implementing rdbmsDbInfoLastBackup because Oracle allows individual tablespaces to be backed up; thus, the time of the last full database backup is not meaningful.

rdbmsDbInfoProductName

Syntax

DisplayString

Max-Access

read-only

Explanation

Displays the textual product name of the server that created or last restructured this database instance. In the Oracle implementation, the value of this variable is retrieved from the BANNER field in V$VERSION.

Typical Range

253 characters

Significance

important

Related Variables

rdbmsSrvProductName

Suggested Presentation

icon or simple string

rdbmsDbInfoVersion

Syntax

DisplayString

Max-Access

read-only

Explanation

Displays the version number of the server that created or last restructured this database instance. The format is specific to the product. In the Oracle implementation, the value of this variable is derived internally.

Typical Range

not applicable

Significance

important (depending on the application)

Related Variables

applVersion

Suggested Presentation

simple string

rdbmsDbInfoSizeUnits

Syntax

INTEGER (1) bytes, (2) kbytes, (3) mbytes, (4) gbytes, or (5) tbytes.

Max-Access

read/write

Explanation

Displays the units used to measure the size of this database instance, as indicated by the values for rdbmsDbInfoSizeAllocated and rdbmsDbInfoSizeUsed. In the Oracle implementation, the value of rdbmsDbInfoSizeUnits is the least that allows rdbmsDbInfoSizeAllocated to be expressed as a 32-bit integer.

(1) bytes, indicates size measured in bytes; (2) kbytes, indicates units of kilobytes; (3) mbytes, indicates units of megabytes; (4) gbytes, indicates units of gigabytes; or (5) tbytes, indicates units of terabytes. Each of these unit measurements are binary multiples (1K = 1024).

Typical Range

enumerated 1 - 5

Significance

very important

Related Variables

rdbmsDbInfoSizeAllocated, rdbmsDbInfoSizeUsed

Suggested Presentation

simple string

rdbmsDbInfoSizeAllocated

Syntax

INTEGER (1 - 2147483647)

Max-Access

read/write

Explanation

Displays the estimated size of this database instance (in rdbmsDbInfoSizeUnits), which is the disk space that has been allocated to it and is no longer available to users on this host. The rdbmsDbInfoSizeAllocated variable does not necessarily indicate the amount of space actually in use for database data; rdbmsDbInfoSizeUsed retrieves this value instead. In the Oracle implementation, the value of this variable is the sum of the BYTES field for the rows of the DBA_DATA_FILES tables.

Typical Range

1 - 2147483647

Significance

very important

Related Variables

rdbmsDbInfoSizeUnits, rdbmsDbInfoSizeUsed

Suggested Presentation

simple string or gauge

rdbmsDbInfoSizeUsed

Syntax

INTEGER (1 - 2147483647)

Max-Access

read-only

Explanation

Displays the estimated size of this database instance (in rdbmsDbInfoSizeUnits), actually in use for database data. In the Oracle implementation, the value of rdbmsDbInfoSizeUsed is the sum of values in the BYTES field of DBA_DATA_FILES minus the sum of values in the BYTES field of DBA_FREE_SPACE. rdbmsDbInfoSizeUsed should always be less than or equal to rdbmsDbInfoSizeAllocated.

Typical Range

1 - 2147483647

Significance

very important

Related Variables

rdbmsDbInfoSizeAllocated, rdbmsDbInfoSizeUnits

Suggested Presentation

gauge

rdbmsSrvTable

The rdbmsSrvTable contains generic information on each database server installed on the managed node. In the standard Oracle configuration, one server supports one database. Thus, while the content of rdbmsSrvTable essentially mirrors the content of rdbmsDbTable, Oracle has chosen to implement both tables in order to fully adhere to one of the central precepts of the public RDBMS MIB standard. This precept is that a distinction between the database and its server must be made, in order to account for other vendors' architectures.

Any installed database that has been configured for SNMP support, whether or not it is actively opened, is represented by a row in the rdbmsDbTable. Each entry in the rdbmsSrvTable represents a given database server on the node. Each entry is indexed by applIndex. The current value of each rdbmsSrvTable variable is retrieved directly from a variety of sources on the managed node.

Oracle implements all rdbmsSrvTable variables. Table D–3 lists each of them, together with its corresponding object ID.

Table D–3 rdbmsDbSrvTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsSrvPrivateMIBOID	1.3.6.1.2.1.39.1.5.1.1

Table D–3 (Cont.) rdbmsDbSrvTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsSrvVendorName	1.3.6.1.2.1.39.1.5.1.2
rdbmsSrvProductName	1.3.6.1.2.1.39.1.5.1.3
rdbmsSrvContact	1.3.6.1.2.1.39.1.5.1.4

rdbmsSrvPrivateMIBOID

Syntax
OBJECT IDENTIFIER

Max-Access
read-only

Explanation
Displays the authoritative identification for the private MIB for this server. In the Oracle implementation, the value of this variable is hardcoded as the root of Oracle's OID subtree (1.3.6.1.4.1.111). If no OBJECT IDENTIFIER exists for the private MIB, attempts to access this object will return noSuchName (SNMPv1) or noSuchInstance (SNMPv2).

Typical Range
1.3.6.1.4.1.111 (only)

Significance
very important

Related Variables
rdbmsDbVendorOID

Suggested Presentation
simple string

rdbmsSrvVendorName

Syntax
DisplayString

Max-Access
read-only

Explanation
Displays the name of the vendor of this installed database server. In the Oracle implementation, the value of this variable is hardcoded in the subagent as "Oracle."

Typical Range

6 characters (only)

Significance

important

Related Variables

rdbmsDbVendorName

Suggested Presentation

icon or simple string

rdbmsSrvProductName

Syntax

DisplayString

Max-Access

read-only

Explanation

Displays the product name of this server. In the Oracle implementation, the value of this variable is hardcoded as “Oracle7 server.”

Typical Range

14 characters (only)

Significance

important

Related Variables

rdbmsDbInfoProductName

Suggested Presentation

icon or simple string

rdbmsSrvContact

Syntax

DisplayString

Max-Access

read/write

Explanation

Displays the textual identification of the contact person for this managed server, together with information indicating how to contact this person. In the Oracle implementation, the value of this variable is retrieved from the SNMP_RW.ORA configuration file

Typical Range

0 - 255 characters

Significance

important

Related Variables

rdbmsDbContact

Suggested Presentation

simple string

rdbmsSrvInfoTable

The rdbmsSrvInfoTable contains additional information about each database server instance actively running on the managed node. The visibility of a database server instance in rdbmsSrvInfoTable is determined by the value of the corresponding applOperStatus object for that database server. (For details about applOperStatus, see Appendix E, "Interpreting Implemented Variables of the Network Services MIB") Monitoring these variables is important for tuning server functions such as I/O, access and activity.

Each entry in the rdbmsDbSrvInfoTable represents a given actively opened database server instance on the node. Each entry is indexed by applIndex. The current value of each rdbmsSrvInfoTable variable is retrieved directly from various V$ dynamic performance tables of this database server instance.

Table D–4 lists each of the rdbmsDbInfoTable variables that Oracle is implementing, together with its corresponding object ID.

Table D–4 rdbmsDbSrvInfoTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsSrvInfoStartupTime	1.3.6.1.2.1.39.1.6.1.1
rdbmsSrvInfoFinishedTransactions	1.3.6.1.2.1.39.1.6.1.2
rdbmsSrvInfoDiskReads	1.3.6.1.2.1.39.1.6.1.3
rdbmsSrvInfoLogicalReads	1.3.6.1.2.1.39.1.6.1.4
rdbmsSrvInfoDiskWrites	1.3.6.1.2.1.39.1.6.1.5
rdbmsSrvInfoLogicalWrites	1.3.6.1.2.1.39.1.6.1.6
rdbmsSrvInfoPageReads	1.3.6.1.2.1.39.1.6.1.7
rdbmsSrvInfoPageWrites	1.3.6.1.2.1.39.1.6.1.8
rdbmsSrvInfoDiskOutOfSpaces*	1.3.6.1.2.1.39.1.6.1.9
rdbmsSrvInfoRequestsHandled	1.3.6.1.2.1.39.1.6.1.10
rdbmsSrvInfoRequestsRecvs*	1.3.6.1.2.1.39.1.6.1.11

Table D–4 (Cont.) rdbmsDbSrvInfoTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsSrvInfoRequestSends*	1.3.6.1.2.1.39.1.6.1.12
rdbmsSrvInfoHighwaterIn-boundAssociations	1.3.6.1.2.1.39.1.6.1.13
rdbmsSrvInfoMaxInboundAssociations	1.3.6.1.2.1.39.1.6.1.14

* Variables marked with an asterisk are not supported. Attempts to access these variables may result in errors.

Oracle is not implementing the following rdbmsDbInfoTable variables:

- rdbmsSrvInfoDiskOutOfSpaces
- rdbmsSrvInfoRequestRecvs
- rdbmsSrvInfoRequestSends

rdbmsSrvInfoStartupTime

Syntax
DateAndTime

Max-Access
read-only

Explanation
Displays the date and time at which this server was last started. In the Oracle implementation, the value of this variable is computed from the STARTUP TIME-JULIAN and STARTUP TIME-SECONDS fields in V$INSTANCE.

Typical Range
not applicable

Significance
important

Related Variables
applUpTime

Suggested Presentation
simple string

rdbmsSrvInfoFinishedTransactions

Syntax
Gauge32

Max-Access

read-only

Explanation

Displays the number of transactions (visible to this server) upon which either COMMIT or ABORT operations have been completed since startup of this database server instance. Some database operations, such as read-only queries, may not result in the creation of a transaction. In the Oracle implementation, the value of this variable is the sum of USER ROLLBACKS and USER COMMITS from V$SYSSTAT.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

none

Suggested Presentation

counter

rdbmsSrvInfoDiskReads

Syntax

Counter32

Max-Access

read-only

Explanation

Displays the total number of reads of database files this server has issued to the operating system since startup of this database server instance. In the Oracle implementation, the value of this variable is retrieved from PHYSICAL READS in V$SYSSTAT.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

rdbmsSrvInfoLogicalReads, rdbmsSrvInfoPageReads

Suggested Presentation

counter

rdbmsSrvInfoLogicalReads

Syntax

Counter32

Max-Access

read-only

Explanation

Displays the total number of logical reads of database files that this server has made internally since startup. This value and the value of rdbmsSrvInfoDiskReads reveal the effect of caching on read operations. In the Oracle implementation, the value of this variable is the sum of CONSISTENT GETS and DB BLOCK GETS in V$SYSSTAT.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

rdbmsSrvInfoDiskReads

Suggested Presentation

counter

rdbmsSrvInfoDiskWrites

Syntax

Counter32

Max-Access

read-only

Explanation

Displays the total number of writes to database files this server has issued to the operating system since startup. In the Oracle implementation, the value of this variable is retrieved from PHYSICAL WRITES in V$SYSSTAT.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

rdbmsSrvInfoPageWrites, rdbmsSrvInfoLogicalWrites

Suggested Presentation

counter

rdbmsSrvInfoLogicalWrites

Syntax

Counter32

Max-Access

read-only

Explanation

The total number of times parts of the database files have been marked “dirty” and in need of writing to the disk. This value and rdbmsSrvInfoDiskWrites give some indication of the effect of “write-behind” strategies in reducing the number of disk writes compared to database operations. Because the writes may be done by servers other than those marking the parts of the database files dirty, these values may only be meaningful when aggregated across all servers sharing a common cache. Numbers are not comparable between products. In the Oracle implementation, the value of this variable is retrieved from DB BLOCK CHANGES in V$SYSSTAT.

Typical Range

0..4294967295

Significance

very important

Related Variables

rdbmsSrvInfoDiskWrites

Suggested Presentation

counter32

rdbmsSrvInfoPageReads

Syntax

Counter32

Max-Access

read-only

Explanation

Displays the total number of pages in database files this server has read since startup of this database server instance. “Pages” are product-specific units of disk I/O operations. This value, together with the value of rdbmsSrvInfoDisksReads, reveals the effect of any grouping read-ahead that may be used to enhance performance of some queries, such as scans. In the Oracle implementation, the value of this variable is the sum of all PHYBLKRD entries in V$FILESTAT.

Typical Range
0 - 4294967295

Significance
very important

Related Variables
rdbmsSrvInfoDiskReads

Suggested Presentation
counter

rdbmsSrvInfoPageWrites

Syntax
Counter32

Max-Access
read-only

Explanation
Displays the total number of pages in database files this server has written since this startup of this database server instance. Pages are product-specific units of disk I/O. This value, together with the value of rdbmsSrvInfoDiskWrites, shows the effect of write strategies that collapse logical writes of continuous pages into single calls to the operating system. In the Oracle implementation, the value of this variable is the sum of all PHYBLKWRT entries in V$FILESTAT.

Typical Range
0 - 4294967295

Significance
very important

Related Variables
rdbmsSrvInfoDiskWrites

Suggested Presentation
counter

rdbmsSrvInfoRequestsHandled

Syntax
Counter32

Max-Access

read-only

Explanation

Displays the total number of requests made to the server on inbound associations since this startup of this database server instance.

In the Oracle implementation, the value of this variable is retrieved from “user calls” in V$SYSSTAT.

Typical Range

0 - 4294967295

Significance

very important

Related Variables

rdbmsSrvInfoRequestRecvs, rdbmsSrvInfoRequestSends

Suggested Presentation

counter

rdbmsSrvInfoHighwaterInboundAssociations

Syntax

Gauge32

Max-Access

read-only

Explanation

Displays the greatest number of inbound associations that have been simultaneously open to this server since this startup of this database server instance. In the Oracle implementation, the value of this variable is retrieved from SESSION_HIGHWATER in V$LICENSE.

Typical Range

0 - value of rdbmsSrvInfoMaxInboundAssociations

Significance

important

Related Variables

rdbmsSrvInfoMaxInboundAssociations

Suggested Presentation

gauge

rdbmsSrvInfoMaxInboundAssociations

Syntax

INTEGER (1..2147483647)

Max-Access

read/write

Explanation

Displays the greatest number of inbound associations that can be simultaneously open with this server. If there is no limit, then the value should be zero. In the Oracle implementation, the value of this variable is retrieved from SESSIONS_MAX in V$LICENSE.

Typical Range

1 - 2147483647

Significance

important

Related Variables

applInboundAssociations, rdbmsSrvInfoHighwaterInboundAssociations

Suggested Presentation

simple string

rdbmsSrvParamTable

The rdbmsSrvParamTable contains configuration parameters for a database server installed on this managed node. This table roughly corresponds to the oraDbConfigTable of the private Oracle Database MIB, covered in Appendix B, "Interpreting Variables of the Oracle Database MIB". If you are developing an Oracle-specific application, you not need to concern yourself with this table; instead, you should focus your attention on oraDbConfigTable, which has been designed for more convenient use.

Each entry in the rdbmsSrvParamTable represents a particular configuration parameter for a database server on the node. Each entry is indexed first by applIndex, then by rdbmsSrvParamName, and finally by rdbmsSrvParamSubIndex; thus, there normally is a one-to-many relationship between a database server and its configuration parameters, and a one-to-many relationship between a configuration parameter and its multiple values. (For details about applIndex, see Appendix E, "Interpreting Implemented Variables of the Network Services MIB") However, in the Oracle implementation, multiple values for a configuration parameter are concatenated into a single entry for that parameter. The current value of each rdbmsSrvInfoTable variable is retrieved directly from a corresponding variable in the INIT.ORA configuration file for this database server.

Entries for a server must be present if the value of the corresponding applOperStatus object is up (1). (See Appendix E, "Interpreting Implemented Variables of the Network Services MIB" for more information on the applOperStatus object.) If an instance of applOperStatus is not up (not 1), then attempts to access corresponding instances in this table may result in either noSuchName (SNMPv1) or noSuchInstance (SNMPv2) being returned by the agent.

Table D–5 lists each rdbmsSrvParamTable variable that Oracle is implementing, together with its corresponding object ID.

Table D–5 rdbmsDbSrvParamTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsSrvParamName	1.3.6.1.2.1.39.1.7.1.1
rdbmsSrvParamSubIndex	1.3.6.1.2.1.39.1.7.1.2
rdbmsSrvParamID*	1.3.6.1.2.1.39.1.7.1.3
rdbmsSrvParamCurrValue	1.3.6.1.2.1.39.1.7.1.4
rdbmsSrvComment*	1.3.6.1.2.1.39.1.7.1.5

* Variables marked with an asterisk are not supported.
Attempts to access these variables may result in errors.

Oracle is not implementing the following rdbmsSrvParamTable variables:

- rdbmsSrvParamComment
- rdbmsSrvParamID

rdbmsSrvParamName

Syntax

DisplayString (1..64)

Max-Access

not-accessible

Explanation

Displays the name of a configuration parameter for a server. This name is product-specific. The length is limited to 64 characters to minimize traffic of keys and to fit on one line. In the Oracle implementation, the value of this variable is the name of a parameter in the INIT.ORA file.

Typical Range

1 - 45 characters

Significance

less important

Related Variables

each oraDbConfigTable variable

Suggested Presentation

simple string

rdbmsSrvParamSubIndex

Syntax

INTEGER (1 - 2147483647)

Max-Access

not-accessible

Explanation

The subindex value for this parameter. In this Oracle implementation, the value of this variable is always 1.

Typical Range

1 (only)

Significance

less important

Related Variables

none

Suggested Presentation

not applicable

rdbmsSrvParamCurrValue

Syntax

DisplayString

Max-Access

read/write

Explanation

Displays the current value for this configuration parameter, as retrieved from the INIT.ORA file in the Oracle implementation.

Typical Range

0 - 255 characters

Significance

less important

Related Variables

oraDbConfigTable variables

Suggested Presentation

simple string

rdbmsRelTable

The rdbmsRelTable indicates how databases and servers on this managed node are related. In the standard Oracle implementation, one server supports one database; nevertheless, this table has been implemented in order to fully adhere to the public RDBMS MIB standard.

Each entry in the rdbmsDbRelTable represents a mapping between a database server and a database on the node. Each entry is indexed first by rdbmsDbIndex, and then by applIndex; thus, for the standard Oracle configuration of one server per database, these two indexes are the same. (For details on applIndex, see Appendix E, "Interpreting Implemented Variables of the Network Services MIB")

Table D–6 lists the rdbmsRelTable variable that Oracle is implementing, together with its corresponding object ID.

Table D–6 rdbmsRelTable Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsRelState	1.3.6.1.2.1.39.1.9.1.1
rdbmsRelActiveTime	1.3.6.1.2.1.39.1.9.1.2

rdbmsRelState

Syntax

INTEGER (2) active or (5) unavailable.

Max-Access

read/write

Explanation

Displays the state of this server's access to this database. Oracle implements (2) active and unavailable (5) only. Active (2) means the server is actively using the database. Unavailable (5) means the database is not available through this server.

Typical Range

2 or 5

Significance

very important

Related Variables

appleOperStatus, rdbmsStateChange

Suggested Presentation

icon or alarm

rdbmsTraps

Unlike other SNMP variables, a trap variable is not an agent's response to a management station's request for information. Rather, it is a message that an agent can send unsolicited to a management station when that agent notices that some condition has occurred (a threshold has been reached, for example).

Table D–7 lists each rdbmsTraps variable and its corresponding object ID.

Table D–7 rdbmsTraps Variables and Corresponding Object IDs

Variable Name	Object ID
rdbmsStateChange	1.3.6.1.2.1.39.2.1.1.1
rdbmsStateChange	1.3.6.1.2.1.39.2.1.1.1

rdbmsStateChange

Objects

{rdbmsRelState}

Explanation

An rdbmsStateChange trap signifies that one of the database servers this on this node has changed its rdbmsRelState in a way that makes it less accessible for use. In the Oracle implementation, active (2) is considered fully accessible. The state sent with the trap is the new, less accessible state ((5) unavailable).

Typical Range

5 (only)

Significance

very important

Related Variables

rdbmsRelState

Suggested Presentation

alarm

E

Interpreting Implemented Variables of the Network Services MIB

This appendix provides information to help you interpret those variables of the public Network Services MIB, Version 2 (RFC 1565) that Oracle has implemented. Specifically, it includes a section covering SNMP variables in the following table:

- applTable

applTable

The applTable contains generic variables designed to apply to all types of network service applications. Certain variables of the applTable have been implemented for the Oracle server. (The Listener does not make use of any Network Services MIB variables.)

Each entry in the applTable represents one of these three applications on the managed node. Each applTable entry for the Oracle server applications represents a database server that has been configured in the SNMP_RW.ORA configuration file, whether or not a current instance of that installed database server is running.

Table E–1 lists the variables of the applTable and indicates which ones have been implemented for the Oracle server.

Table E–1 applTable Variables Implemented for the Oracle Server

Variable Name	Oracle Server
applIndex	x
applName	x
applDirectoryName	
applVersion	x
applUptime	x
applOperStatus	x
applLastChange	x
applInboundAssociations	x
applOutboundAssociations	
applLastInboundActivity	
applLastOutboundActivity	

The only applTable variable that is not implemented for any of these Oracle services is applDirectory, because it only has meaning for sites using OSI directory services.

Table E–2 lists the implemented variables of the applTable and their corresponding object IDs.

A description of all implemented applTable variables follows.

Table E–2 applTable Variables and Corresponding Object IDs

Variable Name	Object ID
applIndex	1.3.6.1.2.1.27.1.1.1
applName	1.3.6.1.2.1.27.1.1.2
applDirectoryName	1.3.6.1.2.1.27.1.1.3
applVersion	1.3.6.1.2.1.27.1.1.4
applUptime	1.3.6.1.2.1.27.1.1.5
applOperStatus	1.3.6.1.2.1.27.1.1.6
applLastChange	1.3.6.1.2.1.27.1.1.7
applInboundAssociations	1.3.6.1.2.1.27.1.1.8
applOutboundAssociations	1.3.6.1.2.1.27.1.1.9
applLastInboundActivity	1.3.6.1.2.1.27.1.1.12
applLastOutboundActivity	1.3.6.1.2.1.27.1.1.13

applIndex

Syntax
INTEGER (0 - 2147483647)

Max-Access
not-accessible

Explanation
An index to uniquely identify the application. This integer is assigned in the SNMP_RW.ORA configuration file

applIndex is implemented for each of the Oracle server services.

Typical Range
0 - 2147483647

Significance
very important

Related Variables
applName

Suggested Presentation
not applicable

applName

Syntax
DisplayString

Max-Access
read-only

Explanation
Indicates the name by which the network service application chooses to be known. applName is implemented for the Oracle server applications.

For the Oracle server, the value of applName is the value of the SID parameter in the SNMP_RO.ORA configuration file.

Typical Range
5 - 25 characters

Significance
very important for the Oracle server

Related Variables
applIndex

Suggested Presentation
simple string

applVersion

Syntax
DisplayString

Max-Access
read-only

Explanation
Indicates the version of the application software. The value is retrieved internally.

applVersion is implemented for each of the Oracle server applications.

Typical Range
1 - 15 characters

Significance
less important

Related Variables
none

Suggested Presentation

simple string

applUptime

Syntax

TimeStamp

Max-Access

read-only

Explanation

Indicates the value of sysUptime, a variable the master agent maintains, which is taken at the time the particular application instance (Oracle server) was last initialized.

If two requests for this variable return the same value, none of the counters associated with this application have been reset between the requests.

applUptime is implemented for the Oracle server.

Typical Range

0..4294967295

Significance

very important

Related Variables

none

Suggested Presentation

gauge or counter

applOperStatus

Syntax

INTEGER (1) up, and (2), down

Max-Access

read-only

Explanation

Indicates the operational status of the application. Oracle is implementing (1) up and (2) down. The UP value (1) indicates that the application is operational and available. The DOWN value (2) indicates that the application is not available. appleOperStatus is implemented for the Oracle server.

If the value of the applOperStatus variable for this application is 2, indicating that it is not available, then the subagent cannot connect to it. In this case, only those MIB table(s) that exist independently of the operational status of that application will be

SNMP-accessible. For the Oracle server, these tables are the rdbmsDbTable and rdbmsSrvTable of the public RDBMS MIB only. All tables of the private Oracle Database MIB would be inaccessible when the value of applOperStatus is 2 (down).

Typical Range

enumerated 1 or 2

Significance

very important

Related Variables

none

Suggested Presentation

alarm or simple string

applLastChange

Syntax

TimeStamp

Max-Access

read-only

Explanation

Indicates the value of sysUptime at the time the network service application entered its current operational state. sysUptime is maintained by the master agent on this managed node. If the current state was entered prior to the last initialization of the local network management subsystem, then this object contains a value of 0.

applLastChange is implemented for the Oracle server.

Typical Range

0 - 4294967295

Significance

less important

Related Variables

applUpTime, sysUptime

Suggested Presentation

simple string

applInboundAssociations

Syntax

Gauge32

Max-Access

read-only

Explanation

Indicates the number of current associations to this application, for which it is the responder. applInboundAssociations is implemented for the Oracle server.

For an Oracle server configured for dynamic single threaded processes, this value is the number of application instances, as retrieved from SESSIONS_CURRENT in V$LICENSE.

Typical Range

1 - 1,000

Significance

less important

Related Variables

none

Suggested Presentation

gauge, ala Tach

F

Interpreting Variables of the Enterprise Manager MIB

This appendix provides information to help you interpret the variables of the private Oracle Enterprise Manager MIB. Specifically, it includes sections covering SNMP variables in the following tables:

- oraEM4AlertTable
- oraEM4JobAlertTable

oraEM4AlertTable

The oraEM4AlertTable describes the SNMP Traps sent from Oracle Enterprise Manager for both metric severity alerts and policy violations.

Table F–1 lists the variables of the oraEM4AlertTable and their corresponding Object IDs.

Table F–1 oraEM4AlertTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraEM4AlertTargetName	1.3.6.1.4.1.111.15.1.1.1.2.1
oraEM4AlertTargetType	1.3.6.1.4.1.111.15.1.1.1.3.1
oraEM4AlertHostName	1.3.6.1.4.1.111.15.1.1.1.4.1
oraEM4AlertMetricName	1.3.6.1.4.1.111.15.1.1.1.5.1
oraEM4AlertKeyName	1.3.6.1.4.1.111.15.1.1.1.6.1
oraEM4AlertKeyValue	1.3.6.1.4.1.111.15.1.1.1.7.1
oraEM4AlertTimeStamp	1.3.6.1.4.1.111.15.1.1.1.8.1
oraEM4AlertSeverity	1.3.6.1.4.1.111.15.1.1.1.9.1
oraEM4AlertMessage	1.3.6.1.4.1.111.15.1.1.1.10.1
oraEM4AlertRuleName	1.3.6.1.4.1.111.15.1.1.1.11.1
oraEM4AlertRuleOwner	1.3.6.1.4.1.111.15.1.1.1.12.1
oraEM4AlertMetricValue	1.3.6.1.4.1.111.15.1.1.1.13.1
oraEM4AlertContext	1.3.6.1.4.1.111.15.1.1.1.14.1

A description of each of these variables follows.

oraEM4AlertTargetName

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The name of the target to which this alert applies.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertTargetType

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The type of the target to which this alert applies.

Typical Range
not applicable

Significance
very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4AlertHostName

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The name of the host on which this alert originated.

Typical Range

not applicable

Significance

very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4AlertMetricName

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The name of the metric or policy which generated this alert.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertKeyName

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The name of the key-column, if present, for the metric which generated this alert.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertKeyValue

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The value of the key-column, if present, for the metric which generated this alert.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertTimeStamp

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The time at which this alert was generated.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertSeverity

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The severity of the alert (for example, Critical).

Typical Range
critical, warning, clear

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertMessage

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The message associated with the alert.

Typical Range
not applicable

Significance
very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4AlertRuleName

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The name of the notification rule that caused this notification.

Typical Range

not applicable

Significance

important

Related Variables

none

Suggested Presentation

not applicable

oraEM4AlertRuleOwner

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The owner of the notification rule that caused this notification.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertMetricValue

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The value of the metric which caused this alert to be generated.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4AlertContext

Syntax
DisplayString

Max-Access
read-only

Status

Mandatory

Explanation

A comma separated list of metric column names and values associated with the metric that caused this alert to be generated.

Typical Range

not applicable

Significance

important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertTable

The oraEM4JobAlertTable describes changes in the status of either a Job or a Corrective Action that is running as part of the Oracle Enterprise Manager Job system.

Table F–2 lists the variables of the oraEM4JobAlertTable and their corresponding Object IDs.

Table F–2 oraEM4JobAlertTable Variables and Corresponding Object IDs

Variable Name	Object ID
oraEM4JobAlertJobName	1.3.6.1.4.1.111.15.1.2.1.2.1
oraEM4JobAlertJobOwner	1.3.6.1.4.1.111.15.1.2.1.3.1
oraEM4JobAlertJobType	1.3.6.1.4.1.111.15.1.2.1.4.1
oraEM4JobAlertJobStatus	1.3.6.1.4.1.111.15.1.2.1.5.1
oraEM4JobAlertTargets	1.3.6.1.4.1.111.15.1.2.1.6.1
oraEM4JobAlertTimeStamp	1.3.6.1.4.1.111.15.1.2.1.7.1
oraEM4JobAlertRuleName	1.3.6.1.4.1.111.15.1.2.1.8.1
oraEM4JobAlertRuleOwner	1.3.6.1.4.1.111.15.1.2.1.9.1
oraEM4JobAlertMetricName	1.3.6.1.4.1.111.15.1.2.1.10.1
oraEM4JobAlertMetricValue	1.3.6.1.4.1.111.15.1.2.1.11.1
oraEM4JobAlertContext	1.3.6.1.4.1.111.15.1.2.1.12.1
oraEM4JobAlertKeyName	1.3.6.1.4.1.111.15.1.2.1.13.1
oraEM4JobAlertKeyValue	1.3.6.1.4.1.111.15.1.2.1.14.1
oraEM4JobAlertSeverity	1.3.6.1.4.1.111.15.1.2.1.15.1

A description of each of these variables follows.

oraEM4JobAlertJobName

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The name of the job to which this alert applies.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4JobAlertJobOwner

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The owner of the job to which this alert applies.

Typical Range
not applicable

Significance
very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertJobType

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The type of the job to which this alert applies.

Typical Range

not applicable

Significance

important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertJobStatus

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The status of the job to which this alert applies.

Typical Range

not applicable

Significance

very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertTargets

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

A comma separated list of target to which this alert applies.

Typical Range

not applicable

Significance

very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertTimeStamp

Syntax

DisplayString

Max-Access

read-only

Status
Mandatory

Explanation
The time at which this job status changed causing this alert.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4JobAlertRuleName

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The name of the notification rule that caused this notification.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4JobAlertRuleOwner

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The owner of the notification rule that caused this notification.

Typical Range
not applicable

Significance
important

Related Variables
none

Suggested Presentation
not applicable

oraEM4JobAlertMetricName

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The name of the metric or policy which caused the Corrective Action to run that caused this alert.

Typical Range
not applicable

Significance
very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertMetricValue

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The value of the metric which caused the Corrective Action to run that caused this alert.

Typical Range

not applicable

Significance

important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertContext

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

A comma separated list of metric column names and values associated with the metric which caused the Corrective Action to run that caused this alert.

Typical Range

not applicable

Significance

important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertKeyName

Syntax

DisplayString

Max-Access

read-only

Status

Mandatory

Explanation

The name of the key-column, if present, for the metric which caused the Corrective Action to run that generated this alert.

Typical Range

not applicable

Significance

very important

Related Variables

none

Suggested Presentation

not applicable

oraEM4JobAlertKeyValue

Syntax

DisplayString

Max-Access

read-only

Status
Mandatory

Explanation
The value of the key-column, if present, for the metric which caused the Corrective Action to run that generated this alert.

Typical Range
not applicable

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

oraEM4JobAlertSeverity

Syntax
DisplayString

Max-Access
read-only

Status
Mandatory

Explanation
The severity of the metric which caused the Corrective Action to run that generated this alert (for example, Critical).

Typical Range
critical, warning, clear

Significance
very important

Related Variables
none

Suggested Presentation
not applicable

Glossary

Agent

A process on a managed node that is responsible for all SNMP transactions on that node. The agent accepts requests from the management station and answers them by returning data. It also initiates traps when exceptional conditions are detected.

ASN.1

Abstract Syntax Notation One. The notation used to define MIB variables.

Control File

An administrative file that is required to start and run a database system. The control file records the physical structure of the database.

DBA_DATA_FILES

A data dictionary view that displays information about database files. Fields in DBA_DATA_FILES are used to retrieve the value of the rdbmsDbInfoSizeAllocated variable and derive the values of the rdbmsDbInfoSizeAllocated and rdbmsDbInfoSizeUsed variable (in Oracle's implementation of the standard RDBMS MIB (RFC 1565)). Also, some of the variables in the oraDbDataFileTable of the private Oracle Database MIB correspond directly to a single column of DBA_DATA_FILES.

DBA_FREE_SPACE

A data dictionary view that displays information about free extents in all tablespaces. DBA_FREE_SPACE is used to derive the value of the rdbmsDbInfoSizeUsed variable (in Oracle's implementation of the public RDBMS MIB (RFC 1565)).

DBA_TABLESPACES

The dynamic performance table containing measures of tablespace activity for an Oracle database since the last instance startup. Values for each variable in the oraDbTablespaceTable of the Oracle Private Database MIB are retrieved directly from a corresponding variable in DBA_TABLESPACES.

INIT.ORA Configuration File

A database system file containing a list of parameters that are read when the database system is started. The parameters identify the database and control system wide resources. By affecting the size of the SGA, they also impact the performance of the instance. Values for each variable in the oraDbConfigTable of the private Oracle Database MIB are retrieved directly from a corresponding parameter in INIT.ORA.

Instance

As defined for the Oracle Server, A system global area (SGA) and the Oracle background processes constitute an Oracle instance. Every time a database is started, a system global area is allocated and Oracle background processes are started. The SGA is deallocated when the instance shuts down.

Library

A collection of one or more PL/SQL program units that are stored together in a file or database, and that can be referenced by several displays at once.

Library Cache

A memory structure containing shared SQL and PL/SQL areas. The library cache is one of three parts of the shared pool.

Listener

The listener is a separate process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server. The Listener is a component of Oracle Net, Oracle's remote data access product.

Managed Node

A platform, such as a UNIX server, on which objects to be monitored or managed reside.

Managed Object

Any entity, device, node, such as a LAN server, gateway, or client application, that can be queried by a management station for its current status.

Management Application

A software application that can be integrated with a management framework to accomplish more specialized network or database tasks.

Management Framework

A software package used to view and react to information retrieved using the SNMP protocol.

Management Station

A node from which objects are monitored or managed using the SNMP protocol. Also known as management console, management system or managing node.

Master Agent

A process that accepts queries from the management framework and communicates with the application in order to answer the query.

MIB

Management Information Base. The collection of variables that can be queried to describe attributes of an object, such as the use, performance, and initialization parameters of a database server, for example.

Network Services MIB

A public MIB (RFC 1565) that contains generic variables designed to apply to all types of network service applications. Oracle has implemented those variables of this MIB that are relevant to the Oracle Server, the Oracle Connection Manager, and the Oracle Net Services products.

Object Identifier (OID)

A sequence of elements, as specified by the SNMP standard (RFC 1442), that uniquely identifies each object. An object identifier (OID) is a sequence of elements that indicates a hierarchical organization of identifiers.

OID Subtree

The portion of a MIB tree that is specific to a given entity. The subtree for the Oracle enterprise is 1.3.6.1.4.1.111. All of the variables in the Oracle MIBs begin with this OID subtree. Also called root OID.

Oracle Connection Manager

A router through which a client connection request may be sent either to its next hop or directly to the database server. Clients who route their connection requests through an Oracle Connection Manager can then take advantage of the session multiplexing, access control, or protocol conversion features configured on that Oracle Connection Manager.

Oracle Database MIB

This private MIB serves to extend the functionality of the public RDBMS MIB. The Oracle Database MIB contains additional RDBMS statistics that are specific to the Oracle server architecture.

Oracle Server

The relational database management systems (RDBMS) developed and sold by Oracle. These RDBMS includes the database that stores information, as well as the server software that accesses and manipulates that information.

Oracle SNMP Peer Encapsulator

Oracle SNMP Peer Encapsulator is a mechanism that can connect any third-party SNMP agents to Oracle SNMP Peer Master Agent.

Oracle SNMP Peer Master Agent

A process that acts as a primary interface between the Network Manager and the subagents. Sometimes, Oracle SNMP Peer Master Agent communicates directly with the elements to be managed rather than communicating with the subagents.

Pin

Execution of an item in the library cache.

RDBMS MIB

A proposed standard MIB for relational databases that has been defined by the IETF Working Group. This MIB allows for database discovery, identification of the database, and characterization of database size and activity level.

Registration

The method by which a subagent, acting on behalf of a managed object, such as a listener or the Oracle server, notifies a master agent of two things: first, that it will accept requests from the master agent for information, and second, the list of items (the MIB) for which it will provide information.

Server

As defined in RFC 1565 (the standard RDBMS MIB) an independent entity that provides access to one or more databases.

Shared Global Area or System Global Area (SGA)

The shared segment containing temporary data caches used by the Oracle server during database operation. The SGA is initialized when the server is started, and unavailable when the server is not running. Also known as System Global Area.

Shared Pool

The portion of the SGA, created on instance startup, that contains shared memory constructs such as shared SQL areas. A shared SQL area is required to process every unique SQL statement submitted to a database. A single shared SQL area is used by multiple applications that issue the same statement, leaving more shared memory for other uses.

SNMP_RO.ORA/SNMP_RW.ORA

Configuration files for a managed node. They contain a list of parameters that are read when an Oracle subagent is started.

Subagent

A process that receives queries for a particular application from the master agent, and sends the answers back to the master agent.

V$ Table

A generic term for a virtual table which allows the user to access memory structures within the SGA. V$ tables can be used to monitor the performance of processes of an Oracle instance. These tables are also called dynamic performance tables, because they are continuously updated while a database is open and in use. The values of many private Oracle Database MIB variables are retrieved directly from corresponding variables in V$ tables.

V$DATABASE

The dynamic performance table containing database information from the control file.

V$FILESTAT

The dynamic performance table containing file read/write statistics for an Oracle database (version 7). Some of the variables in the oraDbDataFileTable of the private Oracle Database MIB are retrieved directly from a single column of V$FILESTAT.

V$INSTANCE

The dynamic performance table containing information on the state of the current instance.

V$LIBRARYCACHE

The dynamic performance table containing measures of all library cache activity since the most recent instance startup. Values for each variable in the oraDbLibraryCacheTable of the private Oracle Database MIB are retrieved directly from a corresponding variable in V$LIBRARYCACHE.

V$LICENSE

The dynamic performance table containing information on license limits.

V$ROWCACHE

The dynamic performance table containing measures of all data dictionary activity since the most recent instance startup.

V$SESSTAT

The dynamic performance table containing the current statistics on total memory used for each current session. In the multi-threaded server architecture, Oracle stores session information in the shared pool, rather than in the memory of user processes. While no variables in the private Oracle Database MIB correspond directly to those in V$SESSTAT, V$SYSSTAT is derived from V$SESSTAT. See also V$SYSSTAT.

V$SGA

The dynamic performance table containing summary information on the System Global Area of an Oracle database (version 7 or version 8). The values of each variable in the oraDbSGATable of the Oracle Private Database MIB are retrieved directly from a corresponding variable in V$SGA.

V$SGASTAT

The dynamic performance table containing detailed information on the System Global Area of an Oracle database (version 7 or 8). Values for each variable in the oraDbSGAStatTable of the private Oracle Database MIB are retrieved directly from a corresponding variable in V$SGASTAT.

V$SYSSTAT

The dynamic performance table containing the current systemwide.

value for each variable in V$SESSTAT. Values for each variable in the oraDbSysTable of the private Oracle Database MIB are retrieved directly from a corresponding variable in V$SYSSTAT.

V$VERSION

The dynamic performance table containing version numbers of core library components in the Oracle Server.

Index